20TE70 "60959001

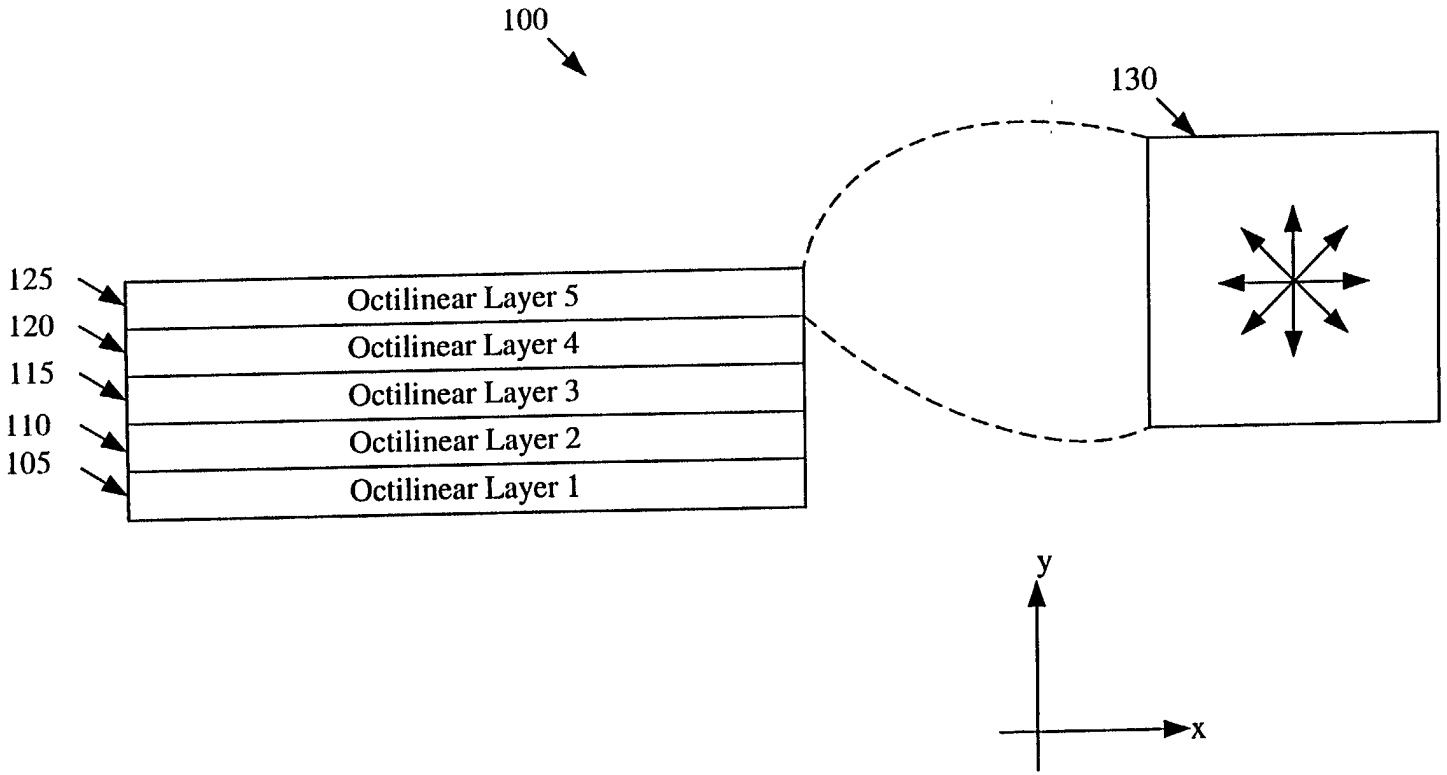

Figure 1

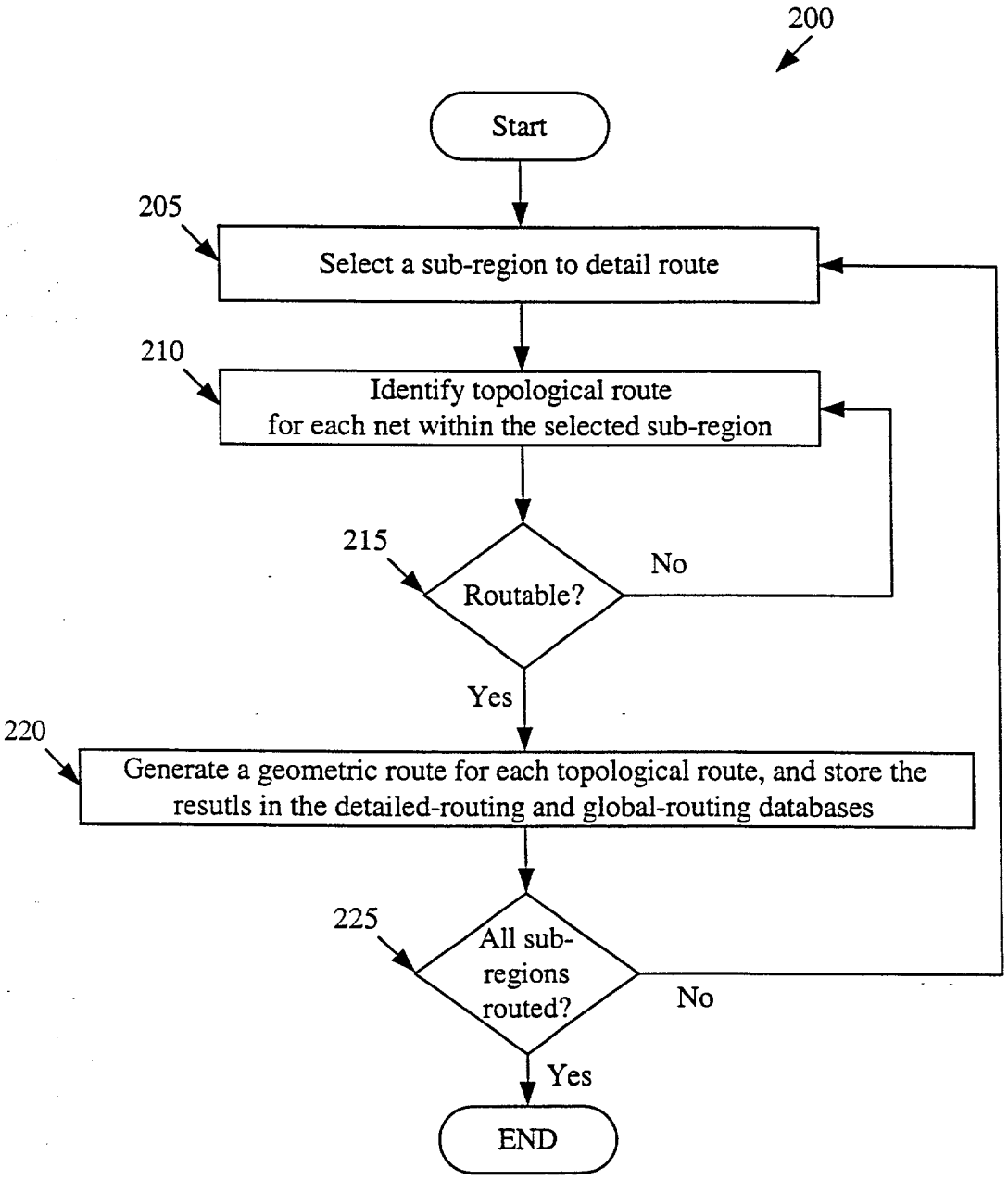
200
Start
205
Select a sub-region to detail route
210
Identify topological route
for each net within the selected sub-region
215
Routable?
No
Yes
220
Generate a geometric route for each topological route, and store the
resutls in the detailed-routing and global-routing databases
225
All sub-
regions
routed?
No
Yes
END

Figure 2

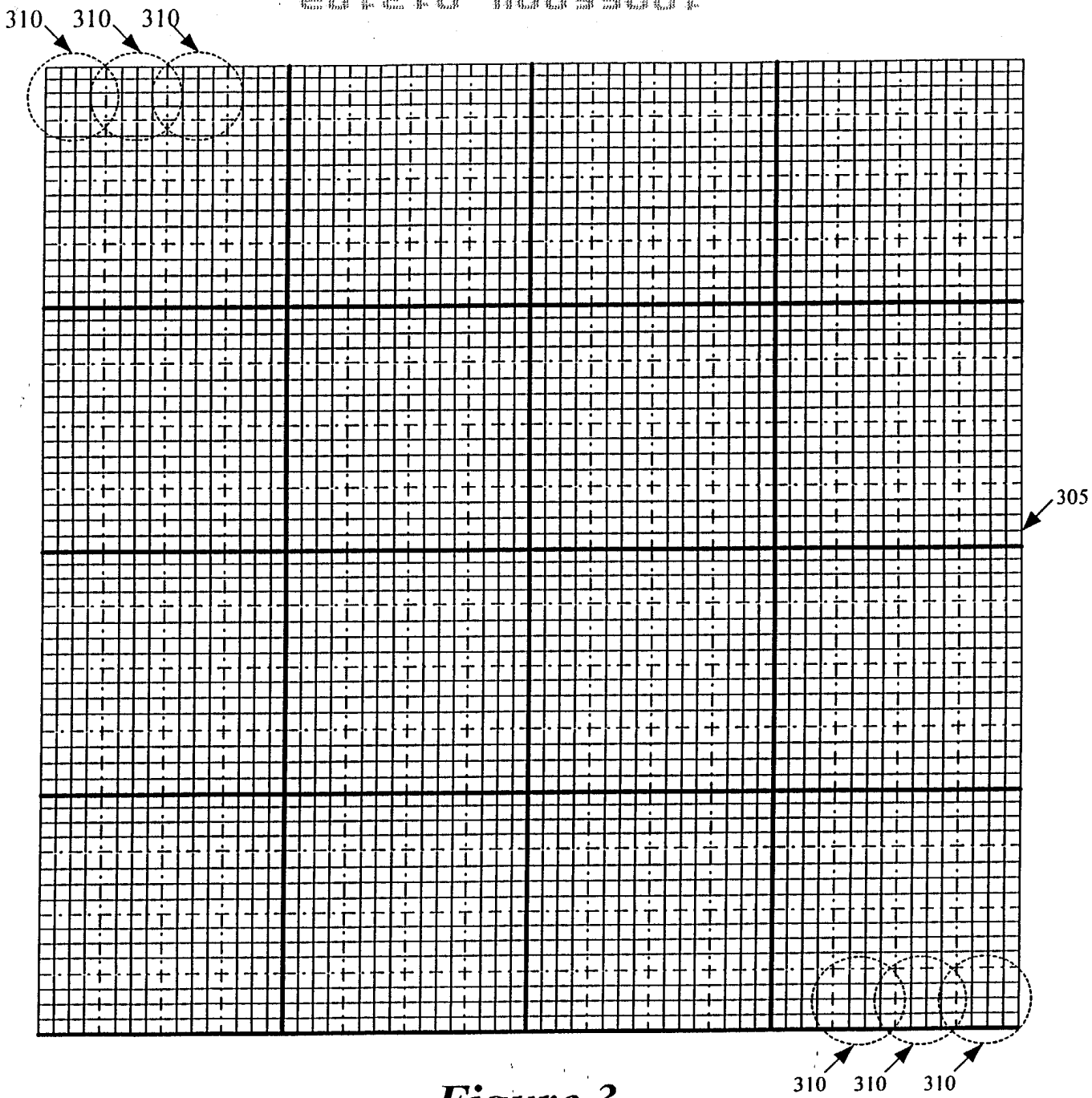

Figure 3

DETECTED: 16098801

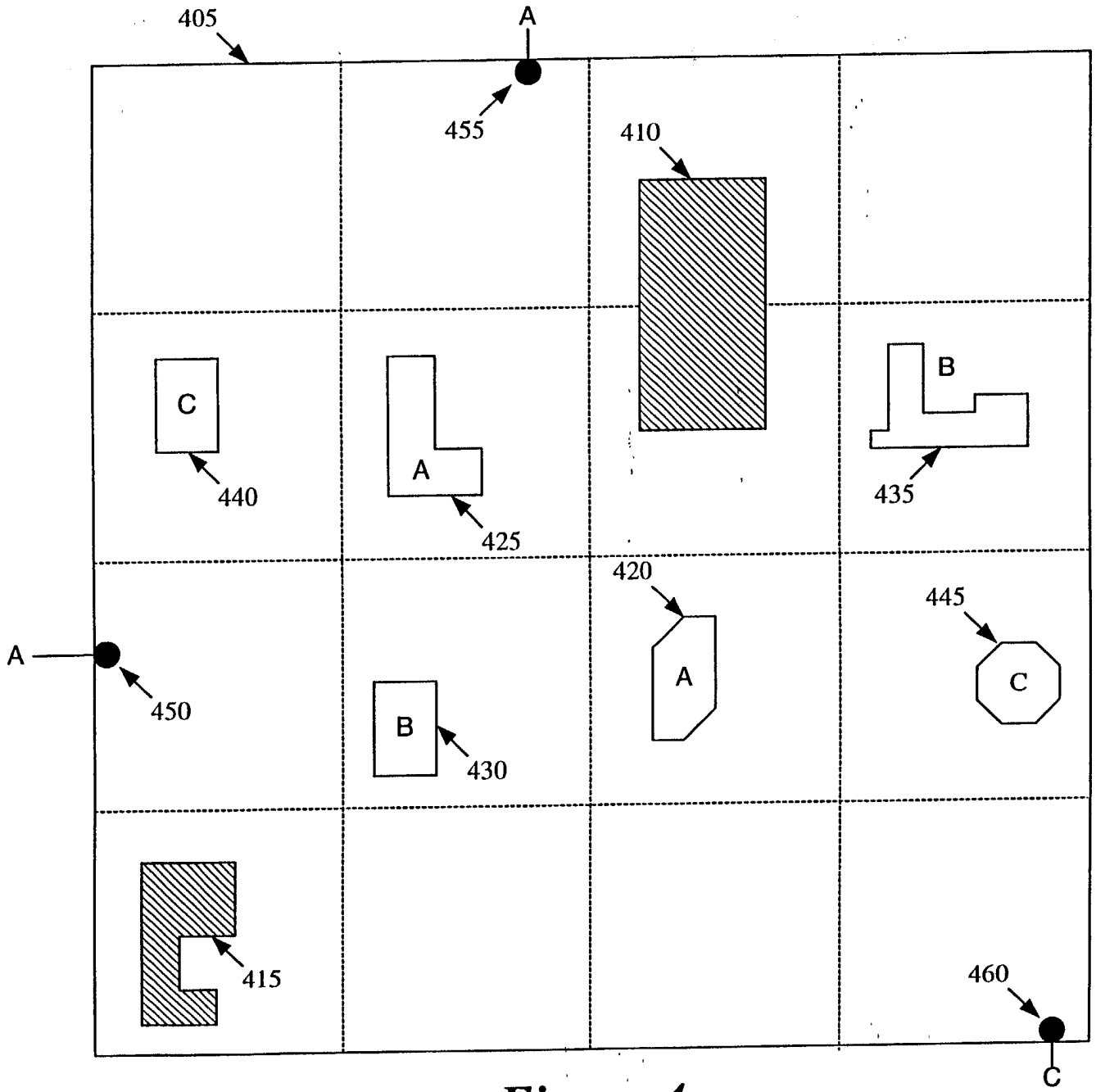

Figure 4

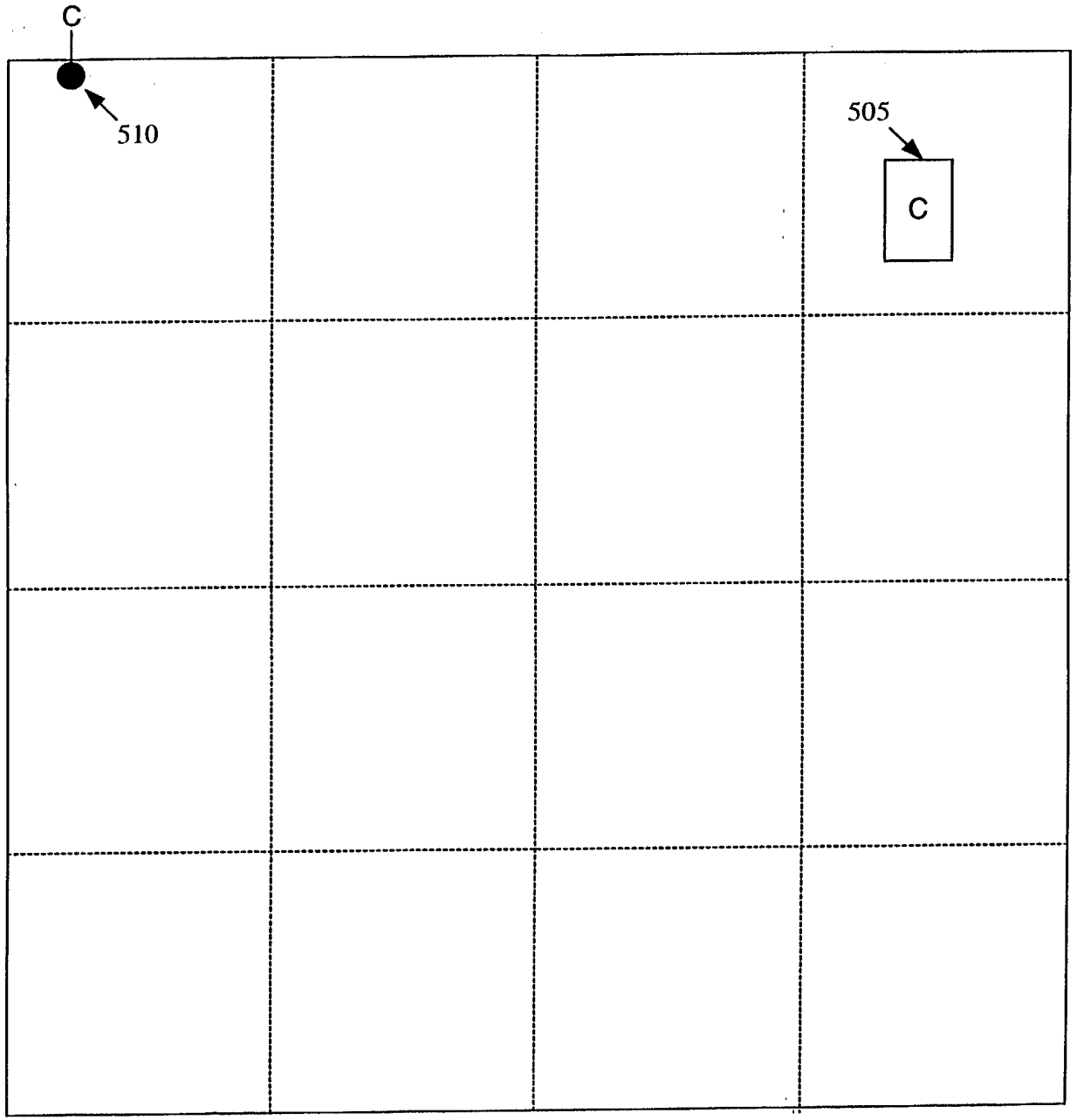

Figure 5

20020 "60990"

-List of Geometries
    --Each Geometry including a sequence of points & layer assignment
-Bounding box of the region
-Array of layer properties
    --Minimum wire size
    --Minimum spacing
    --Via sizes
    --Cost/Unit
-Netlist specifying a number of nets
    --Each net specifying a set of pins
        --Each pin specifying a set of ports
            --Each port specifying a set of geometries

Figure 6

-List of Geometries
    --Each Geometry including a sequence of points & layer assignment
        --List of connection nodes inside each pin geometry
-Bounding box of the region
-Array of layer properties
    --Minimum wire size
    --Minimum spacing
    --Via sizes
    --Cost/Unit
-Netlist specifying a number of nets
    --Each net specifying a set of pins
        --Each pin specifying a set of ports
            --Each port specifying a set of geometries
-For each layer, a graph specifying
    --Nodes
    --Edges
    --Faces

Figure 7

20020091800

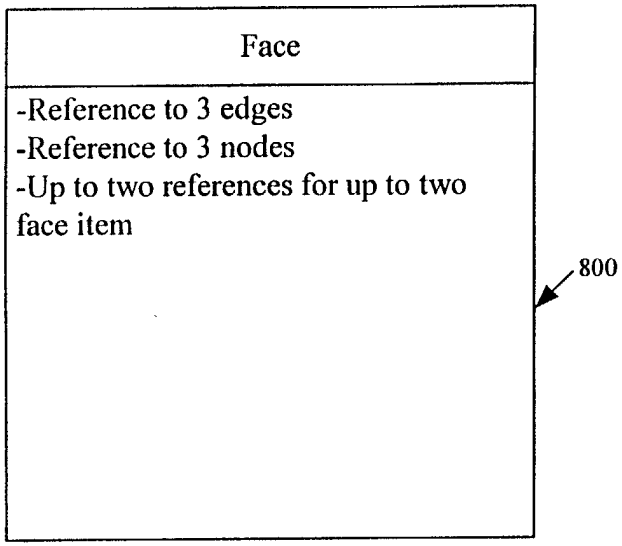

Figure 8

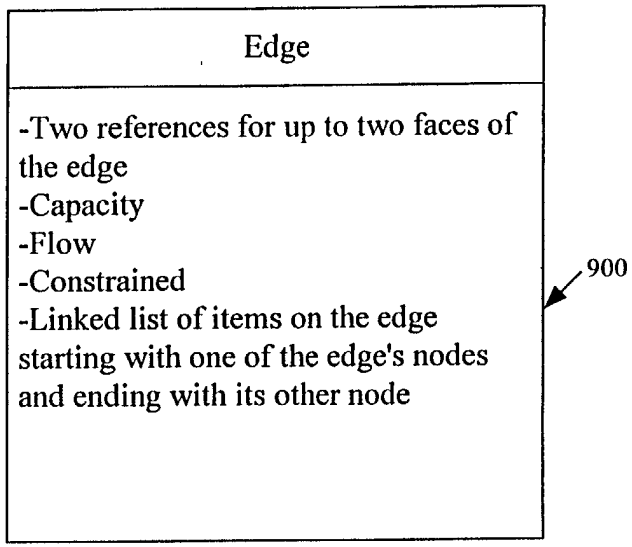

Figure 9

20020704009901

Node
-Net Identifier -One or more planar-path references to adjacent topological items in the same planar path -A pair of via-path references to up and down topological via items -A references to list of edges connected to the node -For each edge, an edge reference to the next or previous topological item on the edge -A reference to the geometry of the node -Vertex number identifying the vertex of the geometry -Location of the node

1000

Figure 10

Edge Item
-Reference to its edge -Net Identifier -A pair of planar-path references to adjacent topological items in the same planar path -A pair of edge references to the next and previous topological item on the edge

1100

Figure 11

Face Item
-Reference to its face -Net Identifier -Up to 3 planar-path references for adjacent topological items in the same planar path -A pair of via-path references for up and down topological via items -Bounding polygon that defines legal face item locations -Constraining Points and Distances

1200

Figure 12

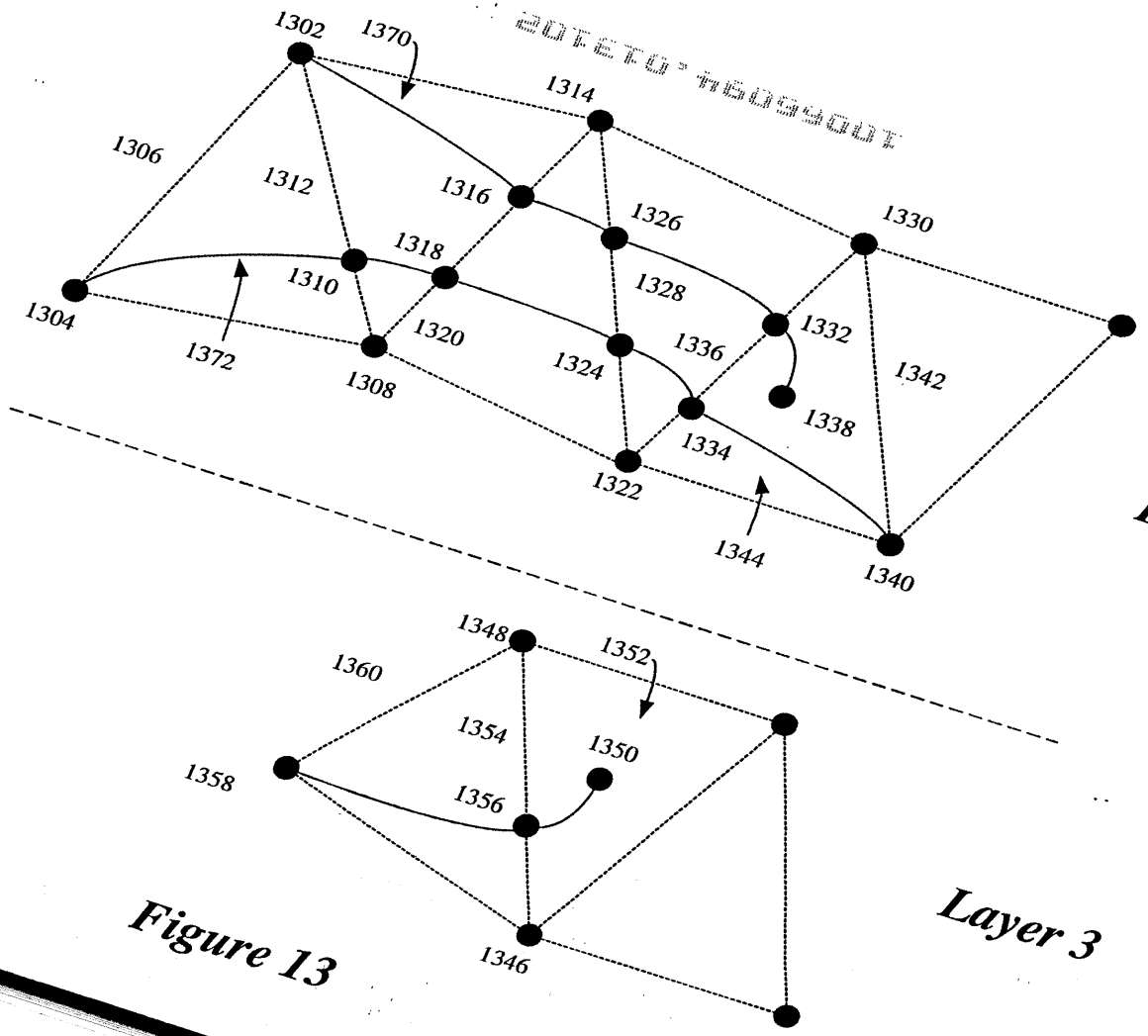
1302
1370
1314
1306
1312
1316
1326
1330
1318
1310
1328
1304
1332
1320
1336
1342
1372
1308
1324
1338
1334
1322
1344
1340
Layer 2
1348
1352
1360
1354
1350
1358
1356
1346
Layer 3

Figure 13

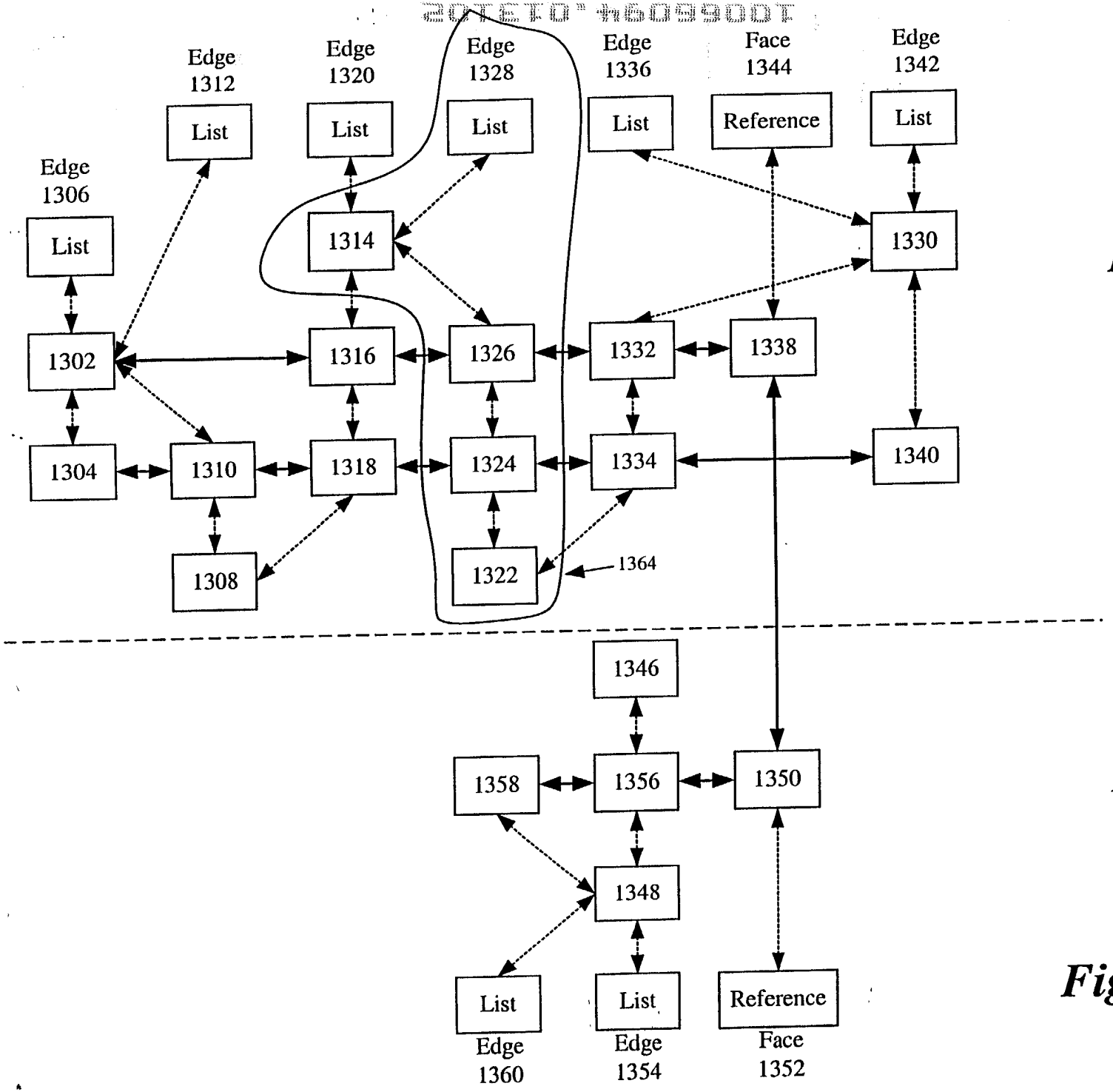
Edge 1312
Edge 1320
Edge 1328
Edge 1336
Face 1344
Edge 1342
Edge 1306
List
Reference
1314
1330
1302
1316
1326
1332
1338
1304
1310
1318
1324
1334
1340
1308
1322
1364
1346
1358
1356
1350
1348
Edge 1360
Edge 1354
Face 1352
Layer 2
Layer 3

Figure 14

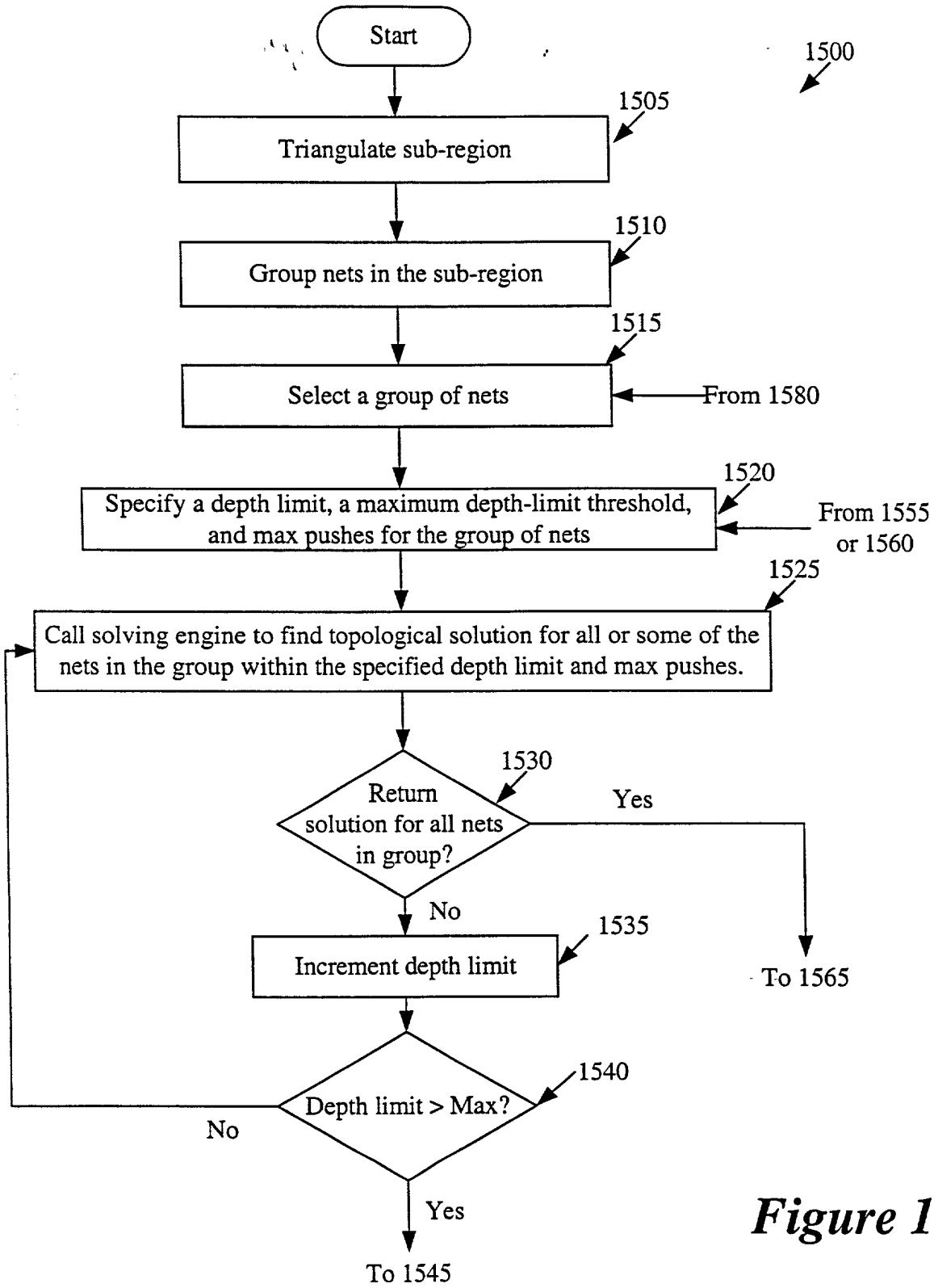

Figure 15A

Figure 15: *Figure 15A* / *Figure 15B*

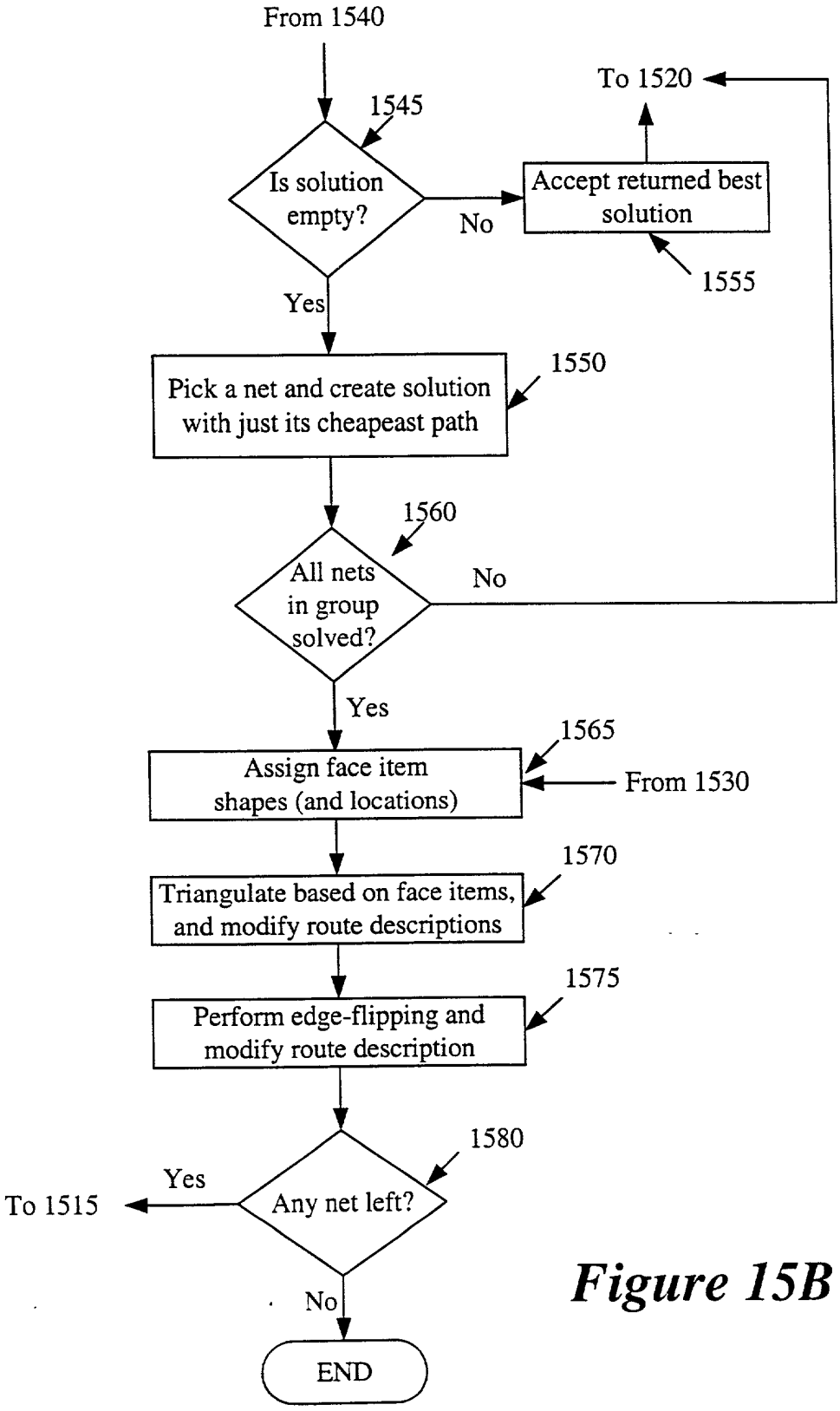
From 1540
1545
Is solution empty?
No
Accept returned best solution
To 1520
1555
Yes
Pick a net and create solution with just its cheapeast path
1550
1560
All nets in group solved?
No
Yes
1565
Assign face item shapes (and locations)
From 1530
1570
Triangulate based on face items, and modify route descriptions
1575
Perform edge-flipping and modify route description
1580
Yes
To 1515
Any net left?
No
END

Figure 15B

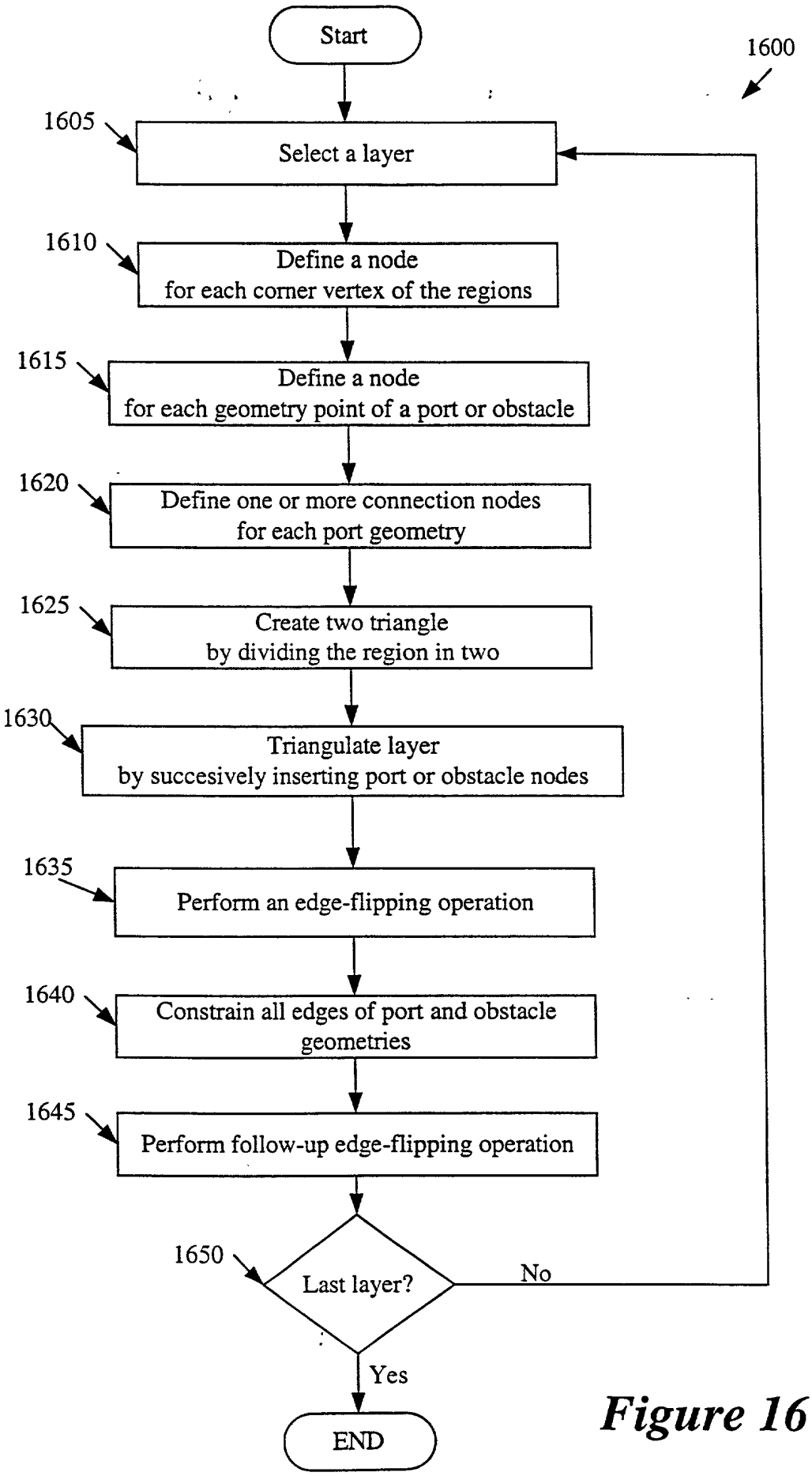
Start
1600
1605
Select a layer
1610
Define a node
for each corner vertex of the regions
1615
Define a node
for each geometry point of a port or obstacle
1620
Define one or more connection nodes
for each port geometry
1625
Create two triangle
by dividing the region in two
1630
Triangulate layer
by succesively inserting port or obstacle nodes
1635
Perform an edge-flipping operation
1640
Constrain all edges of port and obstacle
geometries
1645
Perform follow-up edge-flipping operation
1650
Last layer?
No
Yes
END

Figure 16

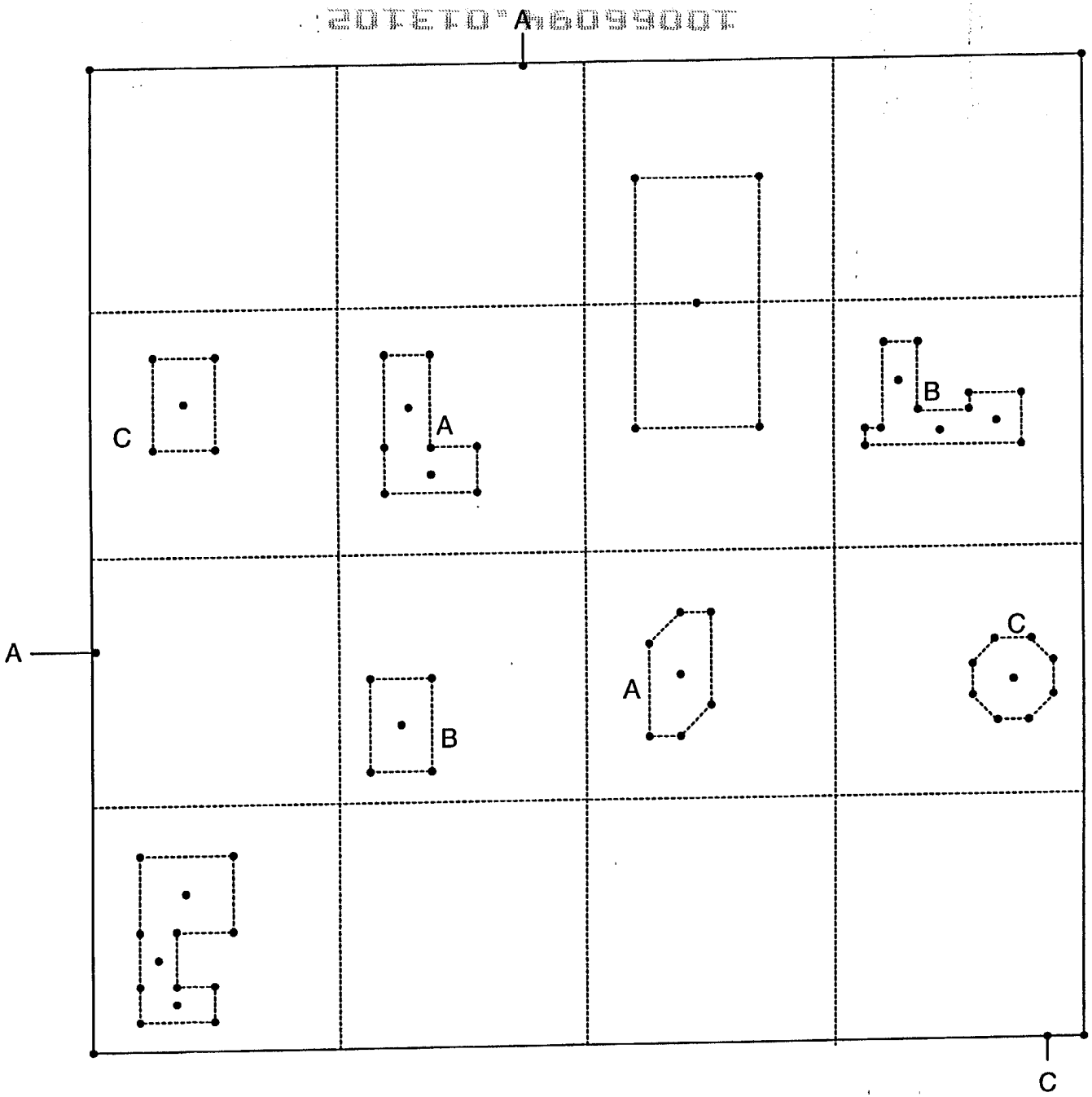

Figure 17

20TETO "t6099001

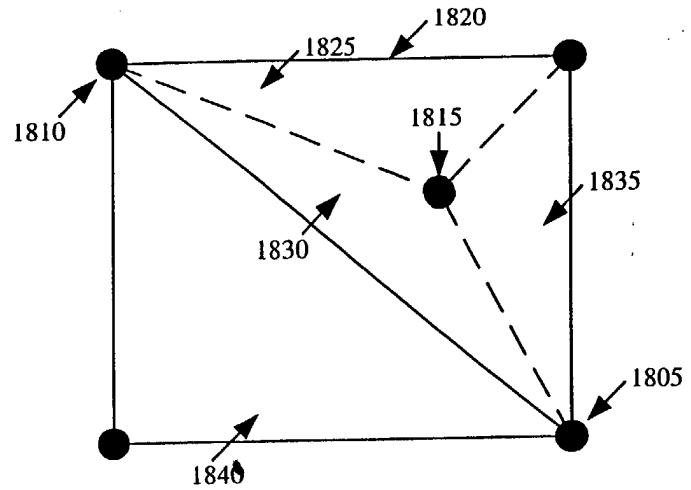

Figure 18

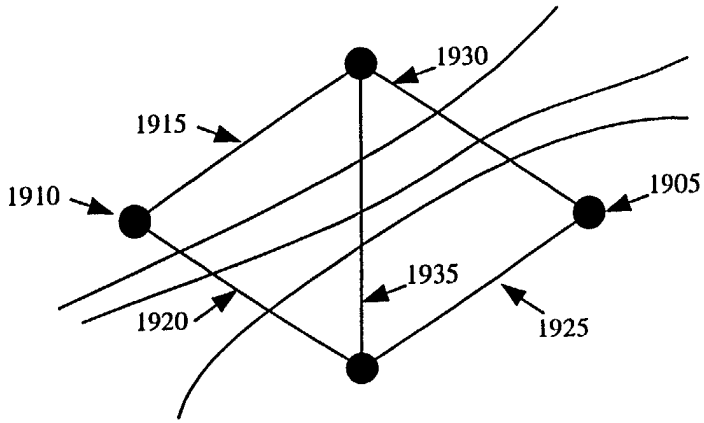

Figure 19

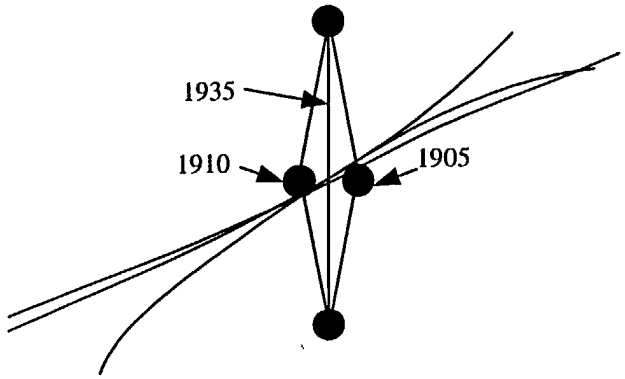

Figure 20

20020804609900A1

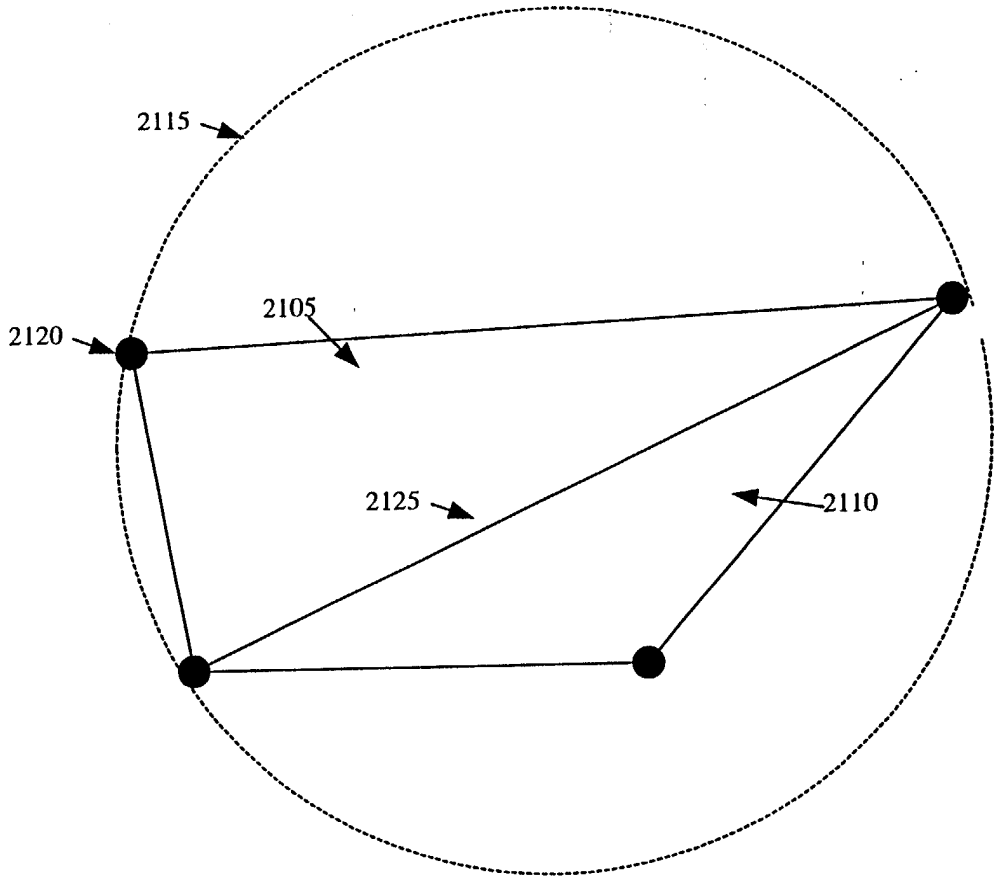

Figure 21

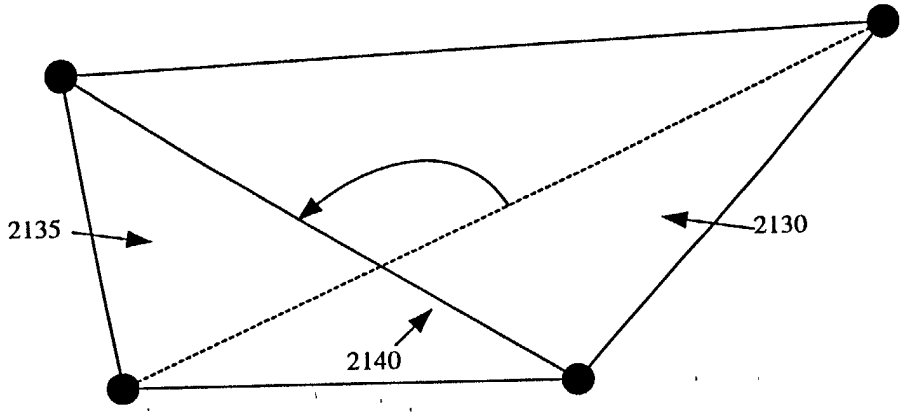

Figure 22

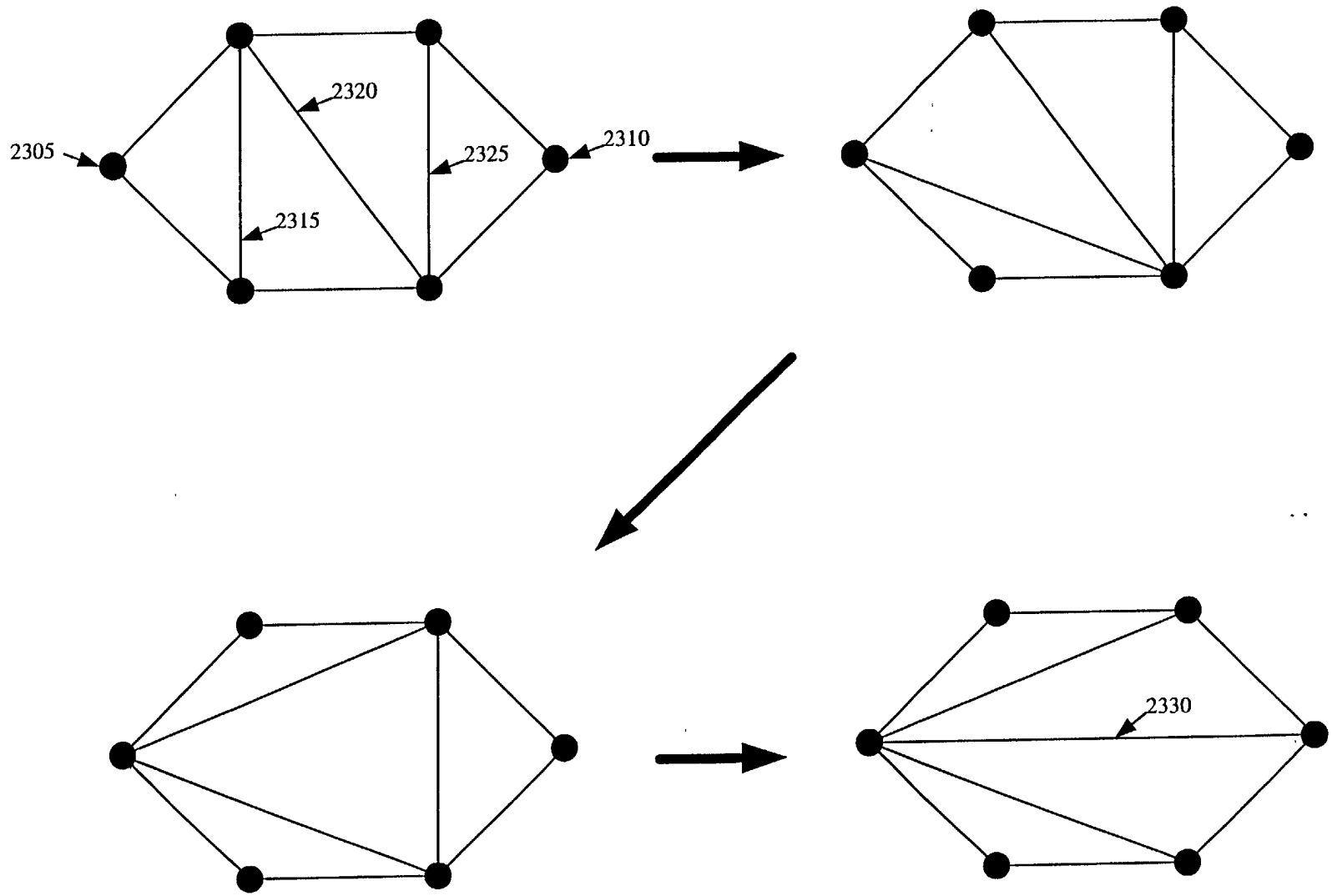

Figure 23

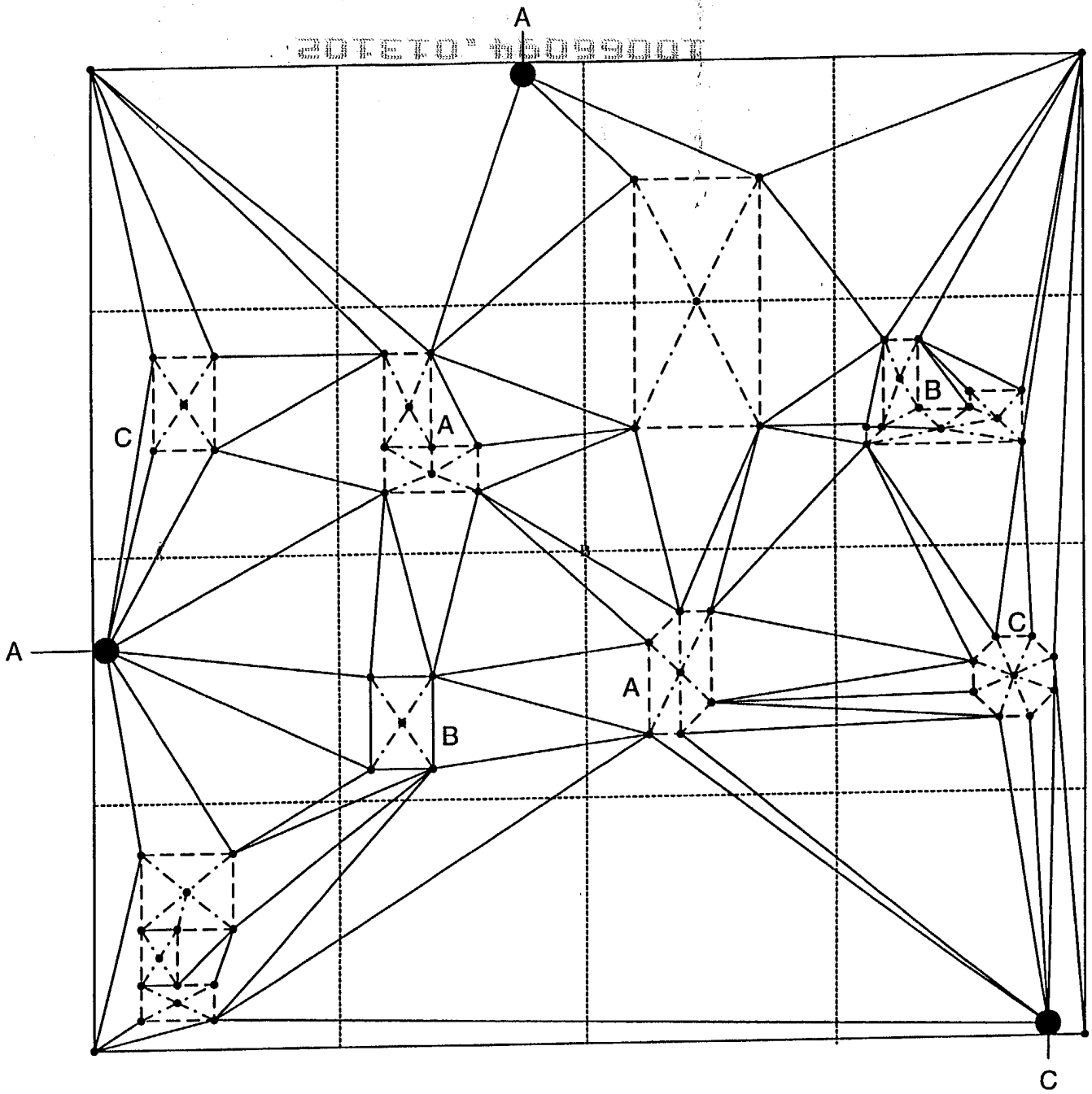

Figure 24

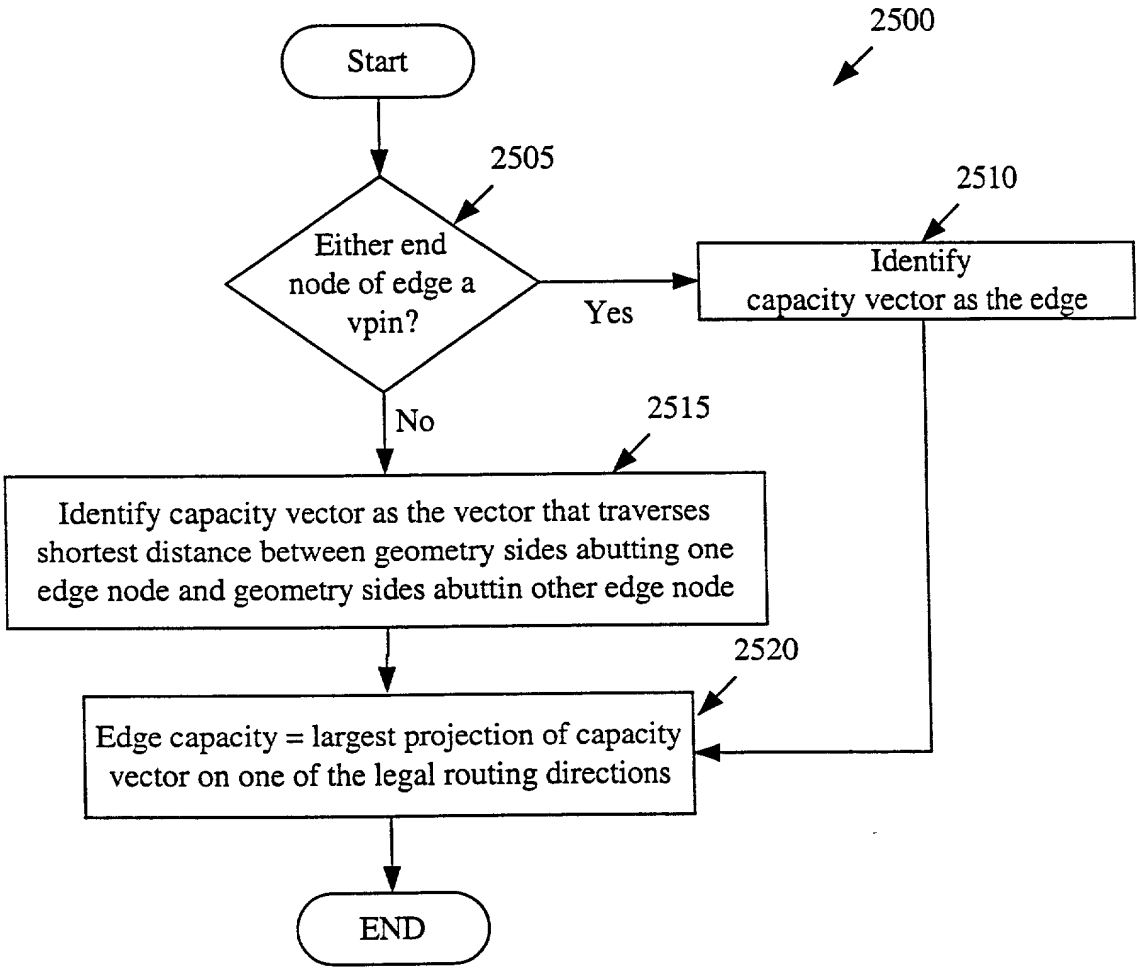
2500
Start
2505
Either end node of edge a vpin?
Yes
2510
Identify capacity vector as the edge
No
2515
Identify capacity vector as the vector that traverses shortest distance between geometry sides abutting one edge node and geometry sides abuttin other edge node
2520
Edge capacity = largest projection of capacity vector on one of the legal routing directions
END

Figure 25

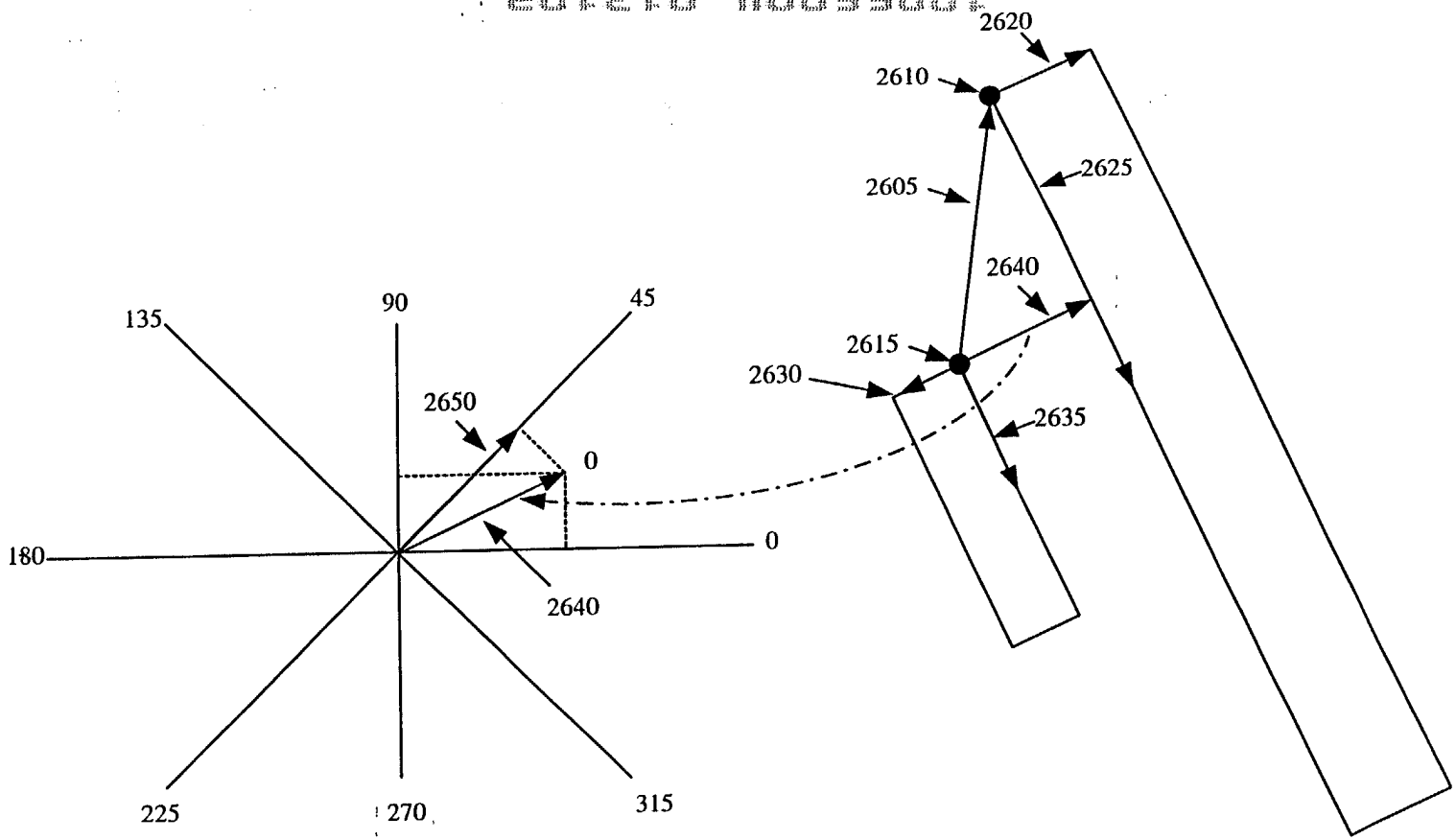
2620
2610
2625
2605
2640
2615
2630
2635
90
45
135
2650
0
180
0
2640
225
270
315

Figure 26

20020704.1609001E

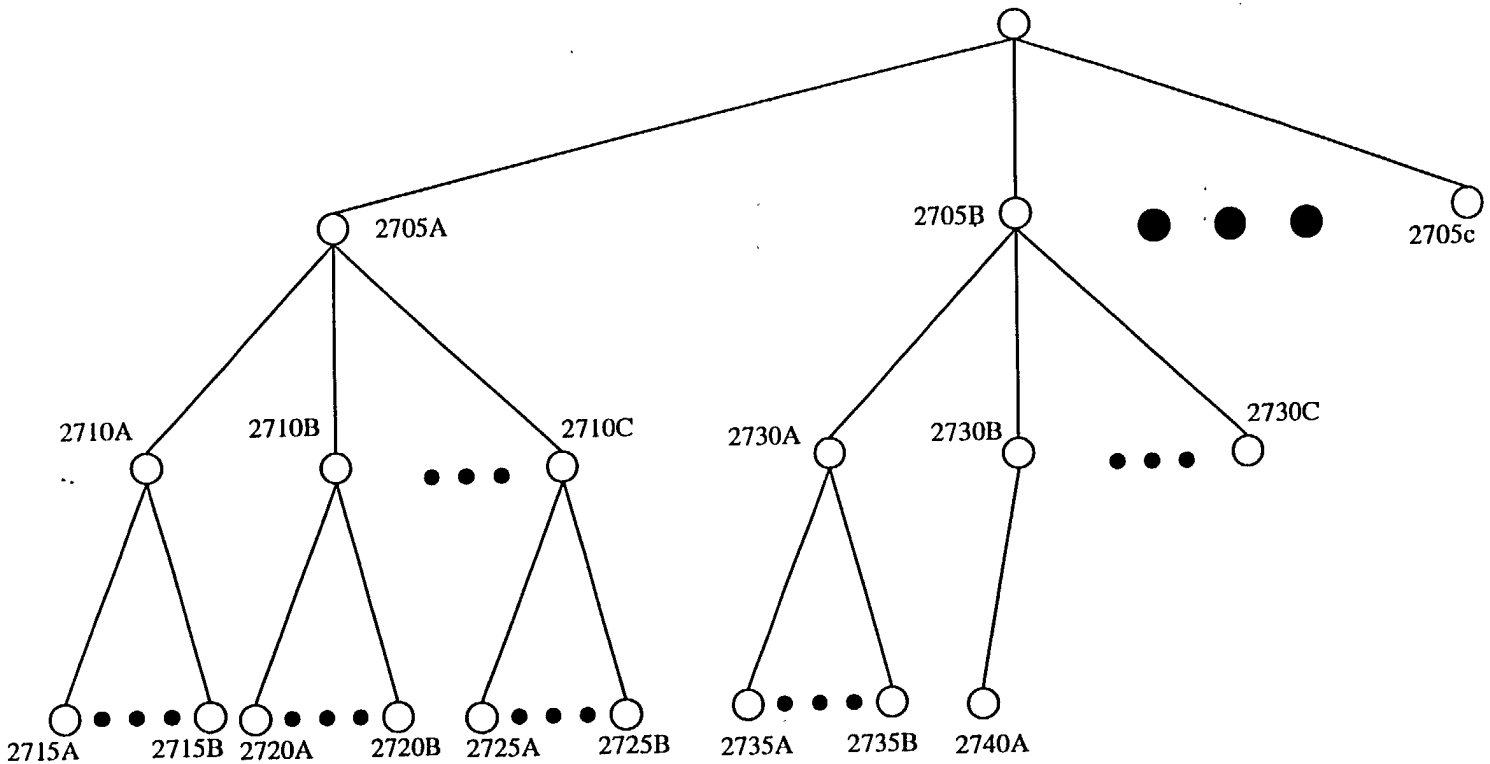

Figure 27

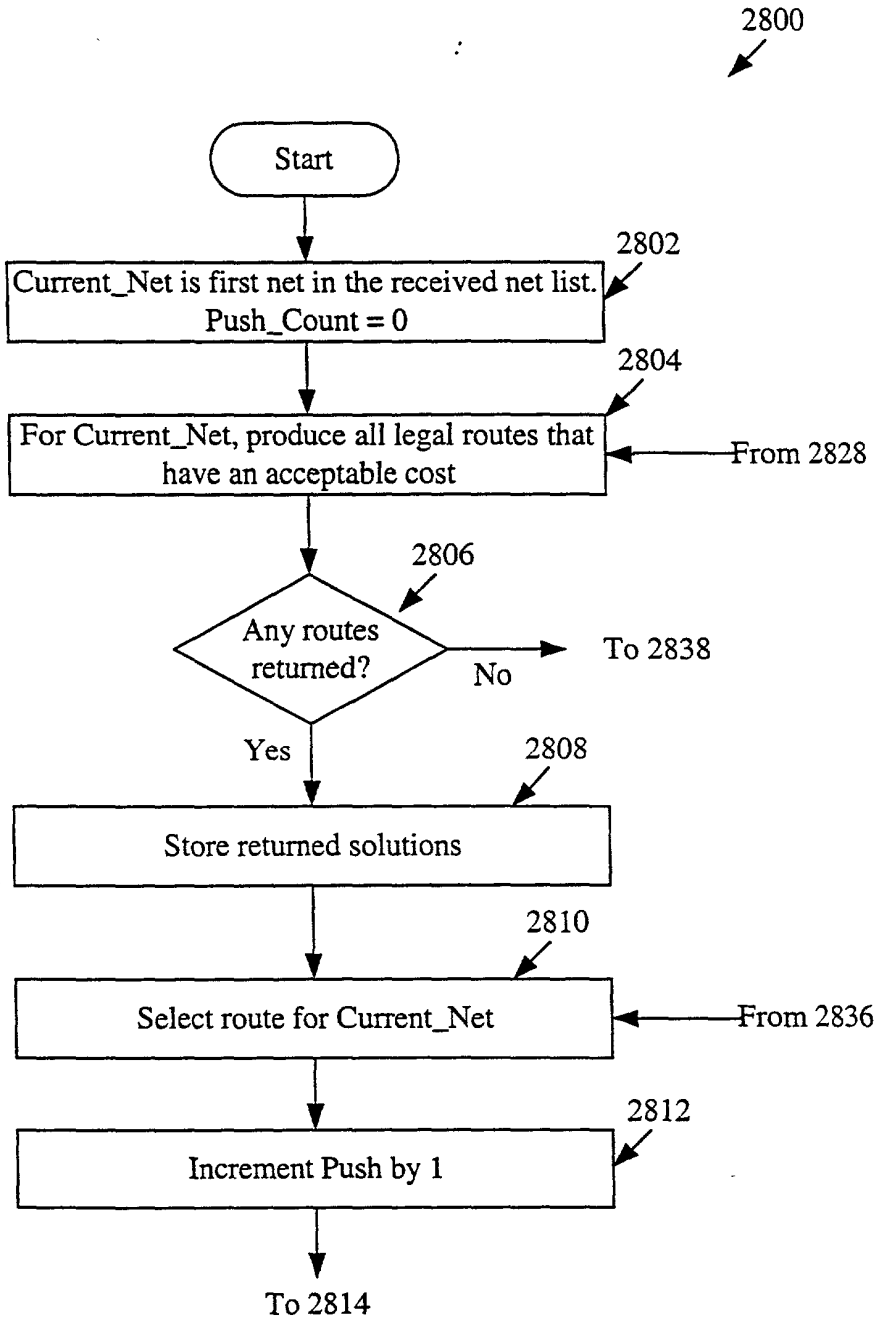

Figure 28A

Figure 28: ***Figure 28A / Figure 28B / Figure 28C***

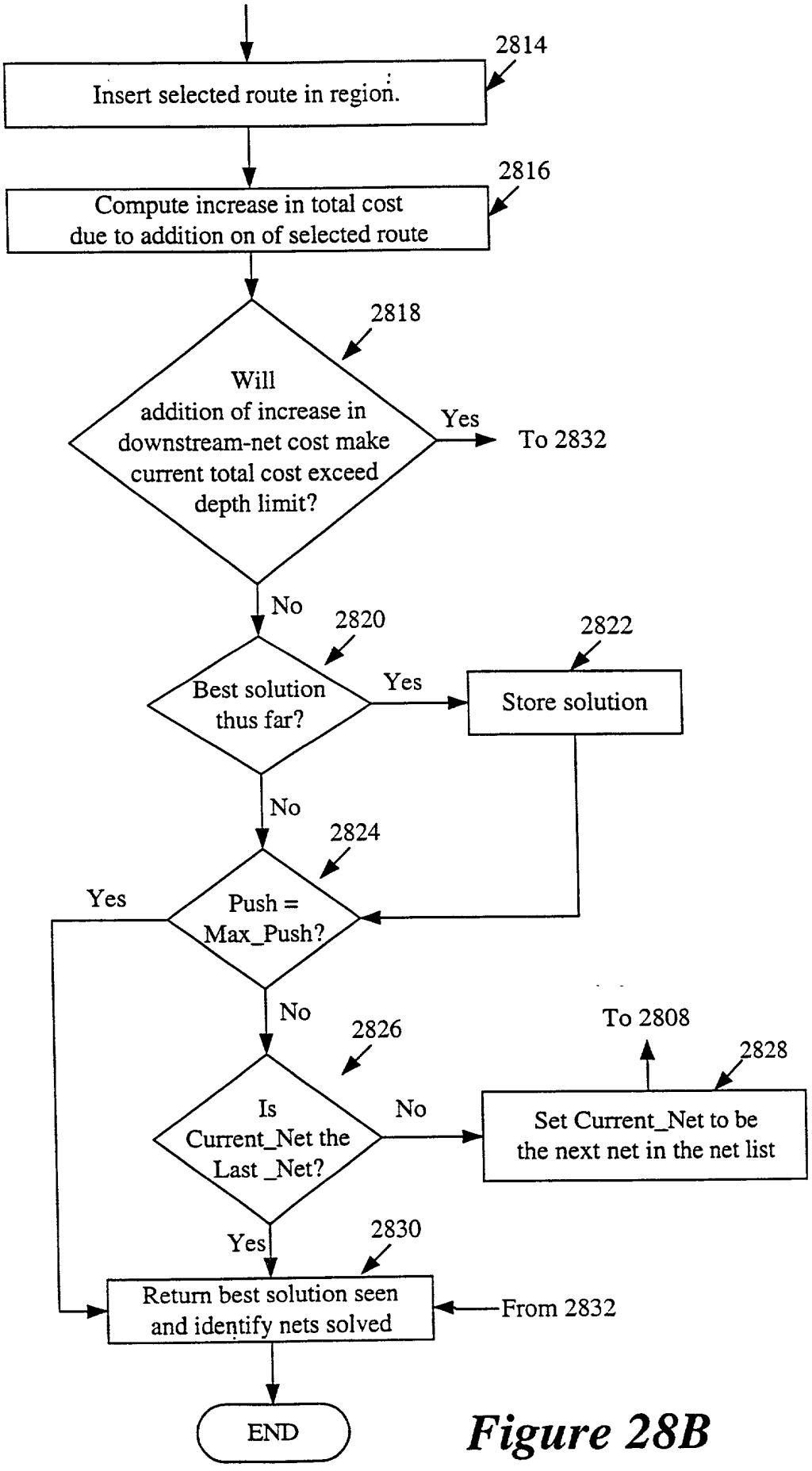
2814
Insert selected route in region.
2816
Compute increase in total cost due to addition on of selected route
2818
Will addition of increase in downstream-net cost make current total cost exceed depth limit?
Yes
To 2832
No
2820
Best solution thus far?
Yes
2822
Store solution
No
2824
Push = Max_Push?
Yes
No
2826
Is Current_Net the Last _Net?
No
To 2808
2828
Set Current_Net to be the next net in the net list
Yes
2830
Return best solution seen and identify nets solved
From 2832
END

Figure 28B

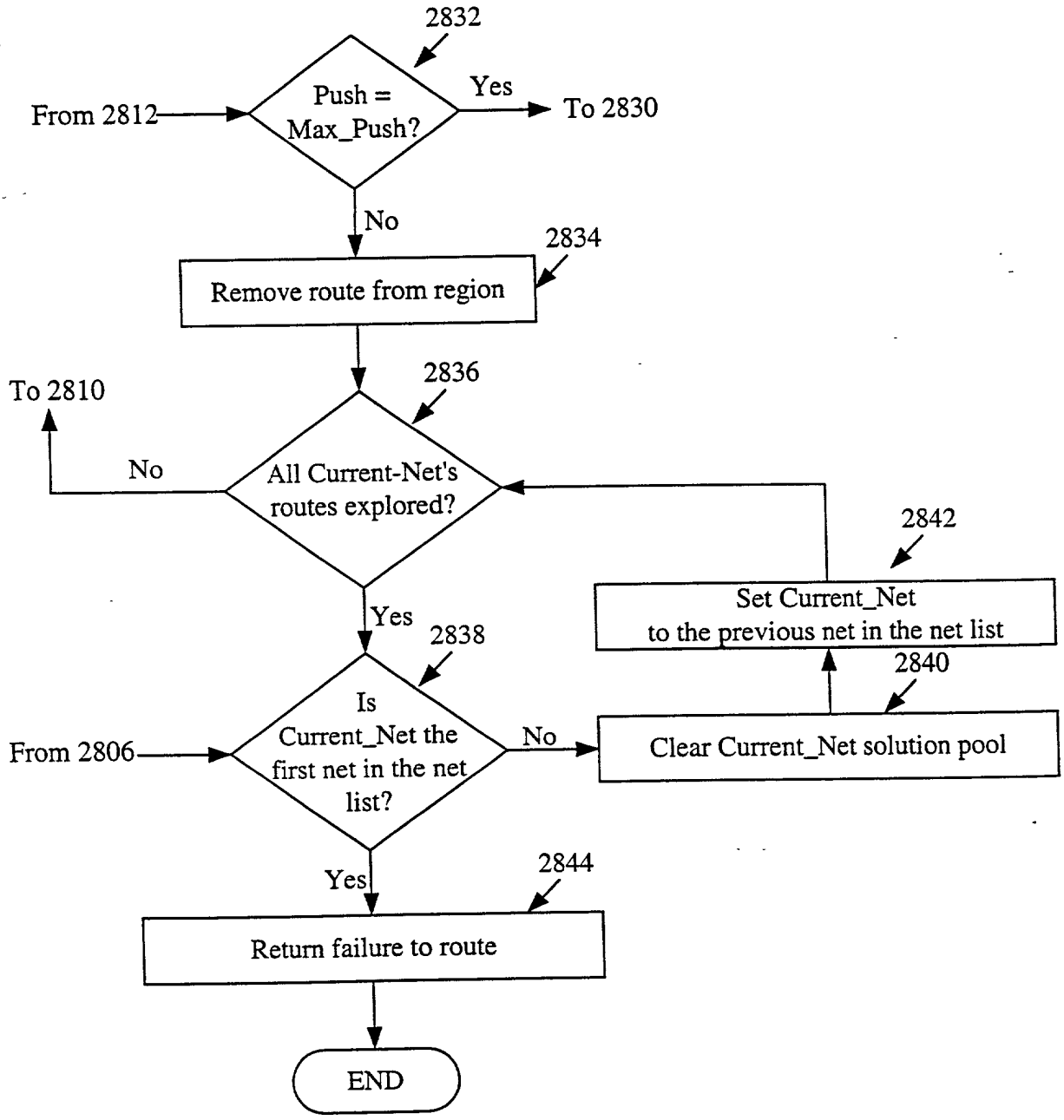
2832
Push = Max_Push?
From 2812
Yes
To 2830
No
2834
Remove route from region
2836
To 2810
No
All Current-Net's routes explored?
2842
Set Current_Net to the previous net in the net list
Yes
2838
2840
Is Current_Net the first net in the net list?
From 2806
No
Clear Current_Net solution pool
Yes
2844
Return failure to route
END

Figure 28C

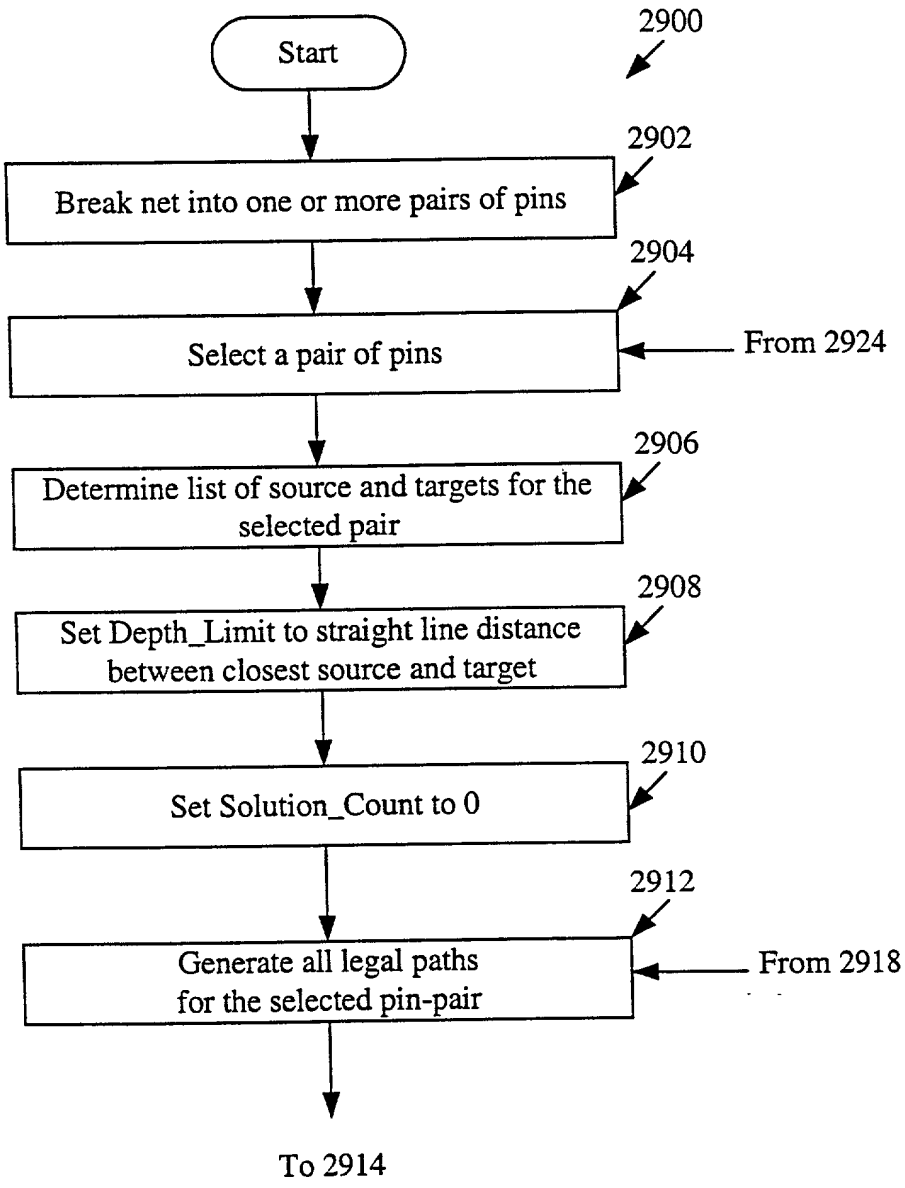

Figure 29A

Figure 29: Figure 29A / Figure 29B

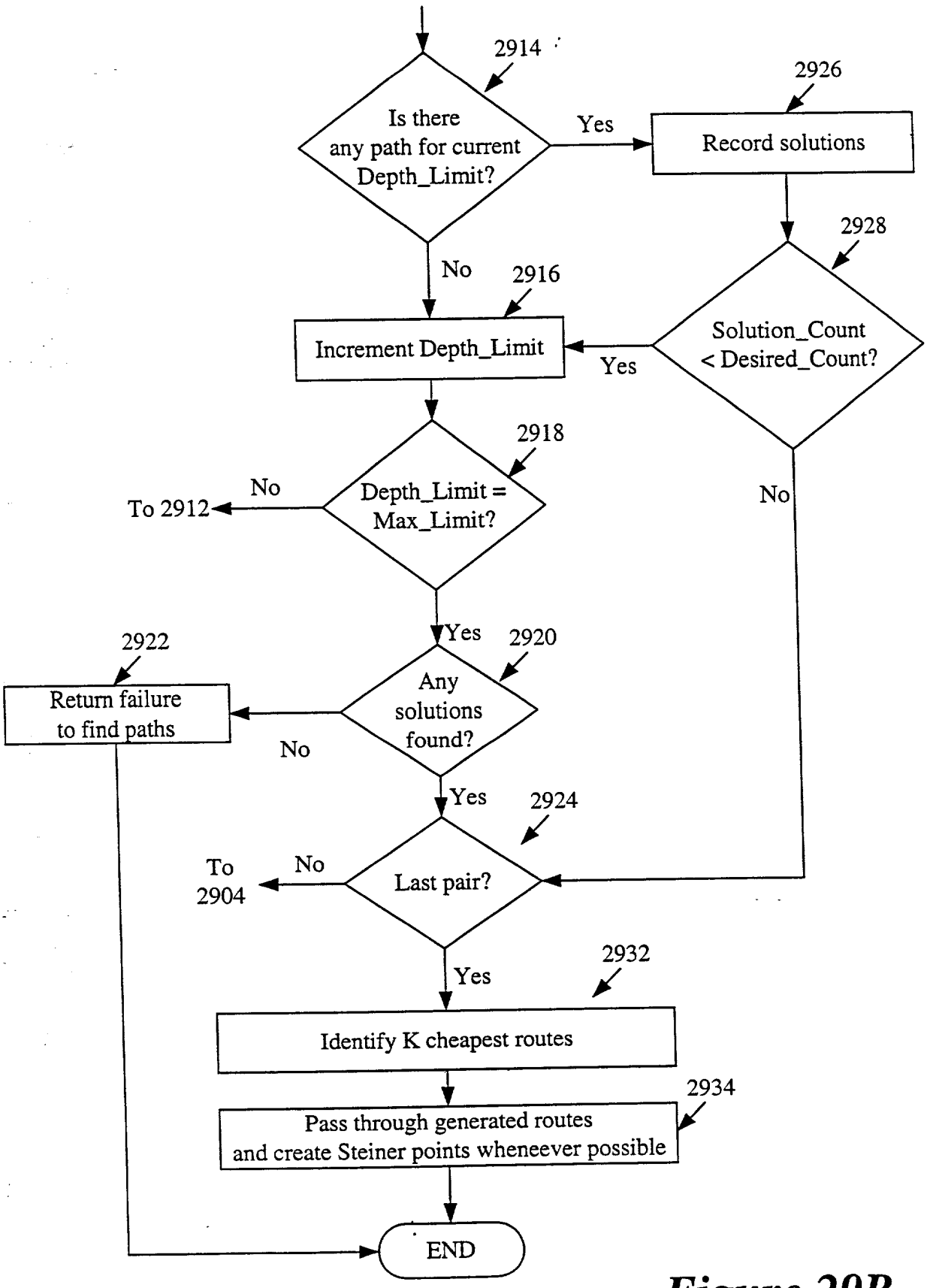
2914
Is there any path for current Depth_Limit?
Yes
2926
Record solutions
2928
Solution_Count < Desired_Count?
Yes
No
2916
Increment Depth_Limit
2918
Depth_Limit = Max_Limit?
No
To 2912
No
Yes
2920
Any solutions found?
2922
Return failure to find paths
No
Yes
2924
Last pair?
To 2904
No
Yes
2932
Identify K cheapest routes
2934
Pass through generated routes and create Steiner points wheneever possible
END

Figure 29B

20TETO"46099DOT

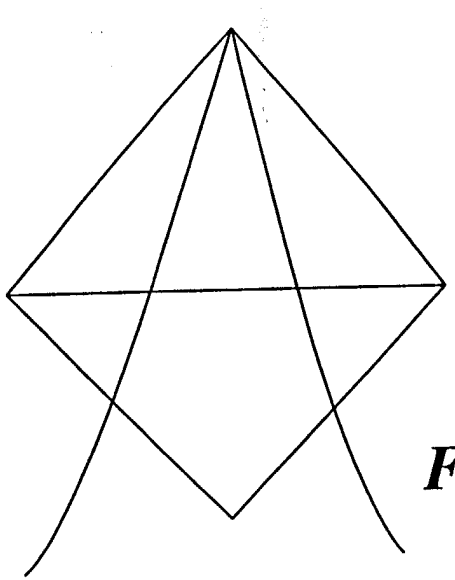

Figure 30A

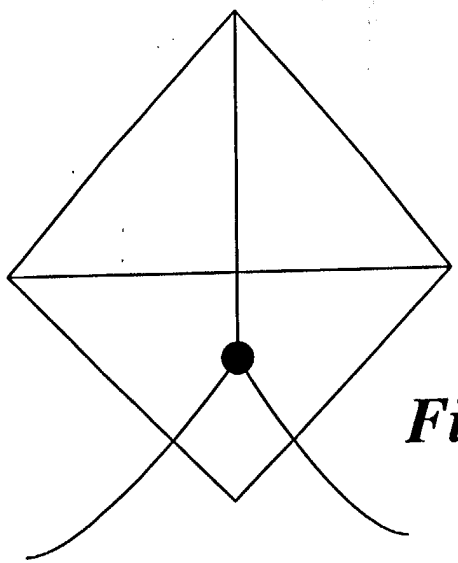

Figure 30B

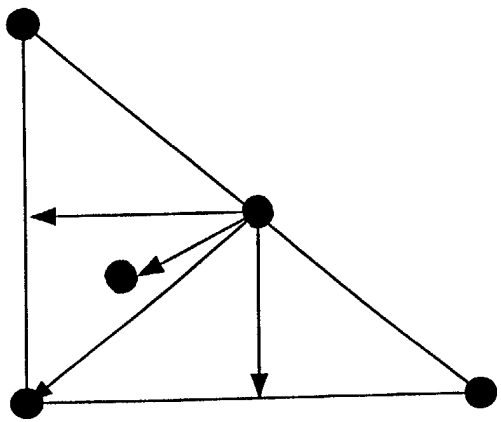

Figure 32

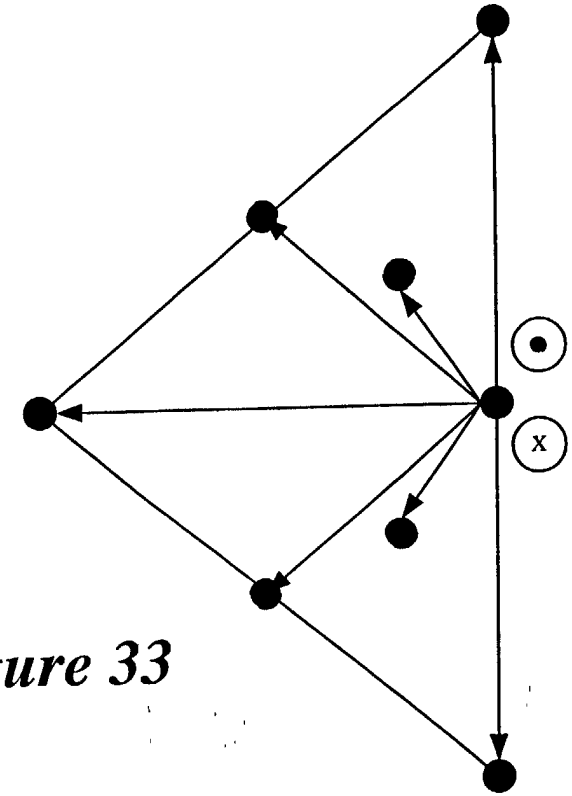

Figure 33

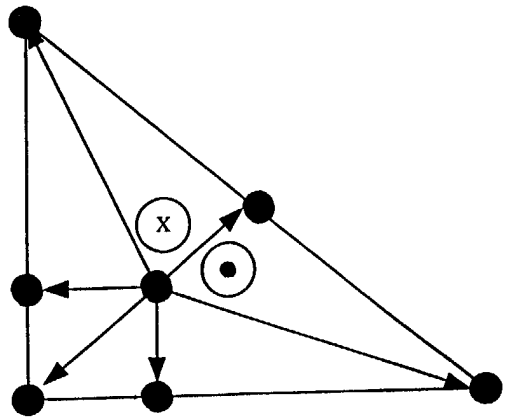

Figure 34

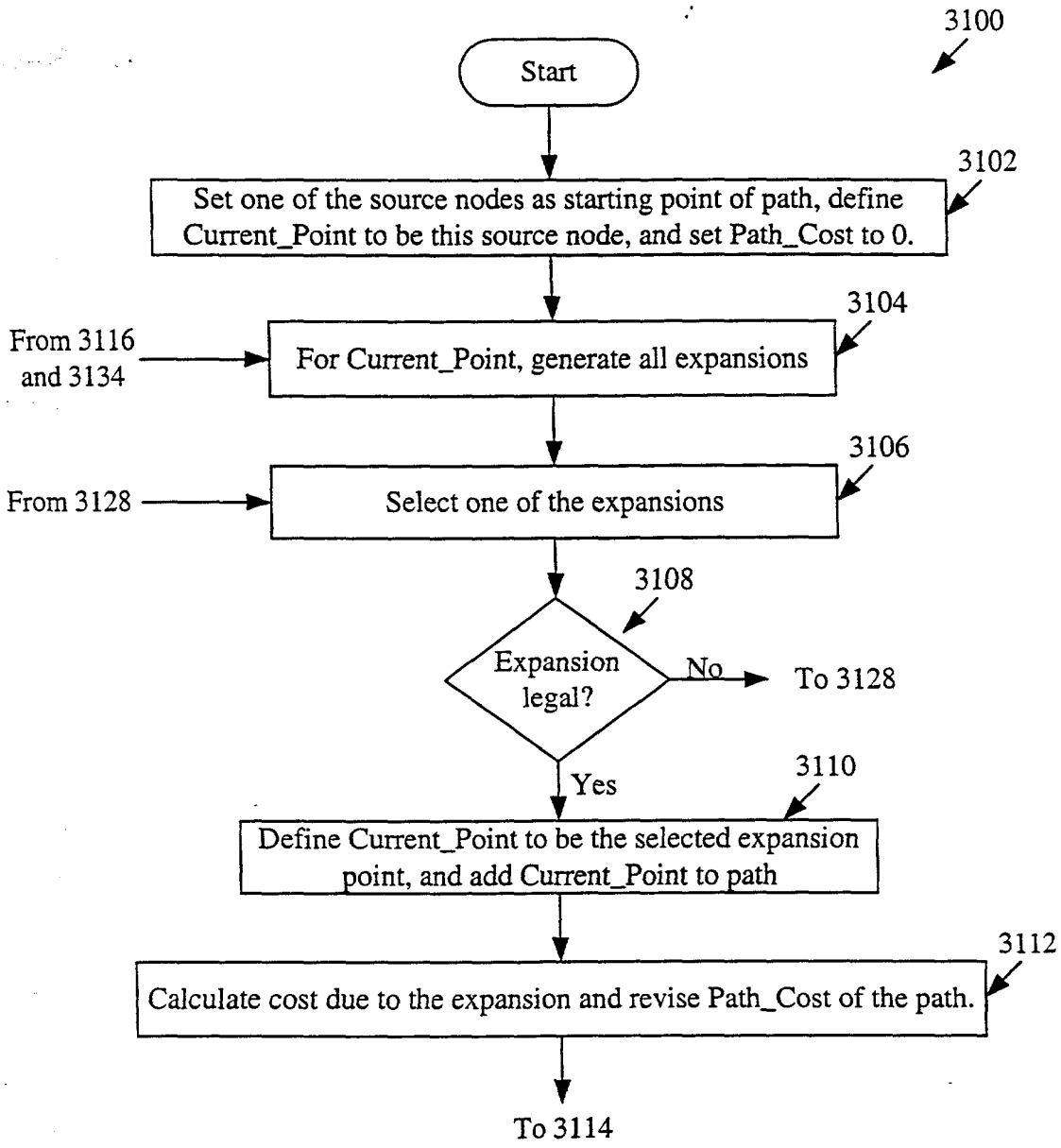

Figure 31A

Figure 31: Figure 31A / Figure 31B

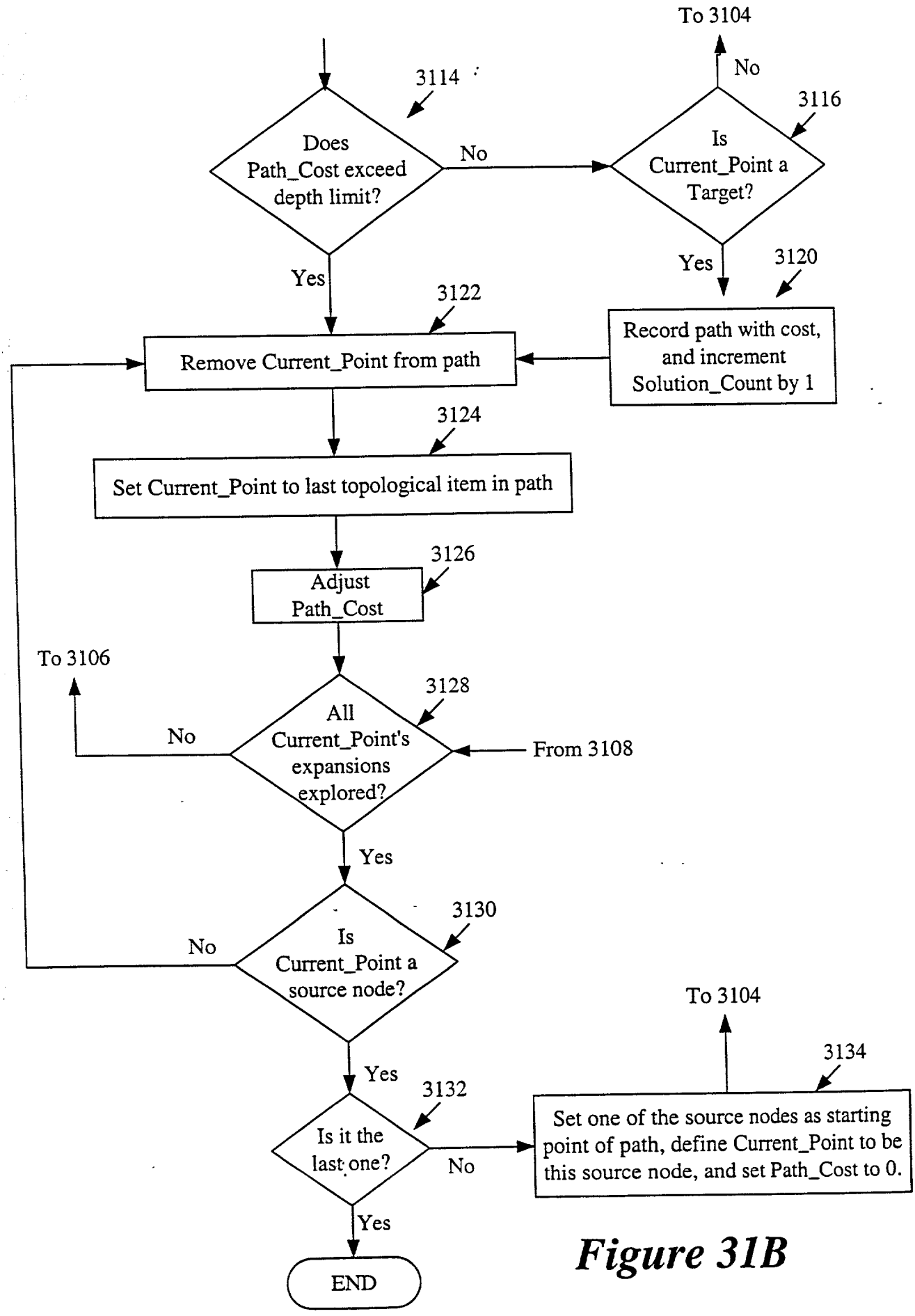
To 3104
No
3114
3116
Does Path_Cost exceed depth limit?
No
Is Current_Point a Target?
Yes
3120
Yes
3122
Record path with cost, and increment Solution_Count by 1
Remove Current_Point from path
3124
Set Current_Point to last topological item in path
3126
Adjust Path_Cost
To 3106
3128
All Current_Point's expansions explored?
No
From 3108
Yes
3130
Is Current_Point a source node?
No
To 3104
3134
Yes
3132
Set one of the source nodes as starting point of path, define Current_Point to be this source node, and set Path_Cost to 0.
Is it the last one?
No
Yes
END

Figure 31B

20120704609900T

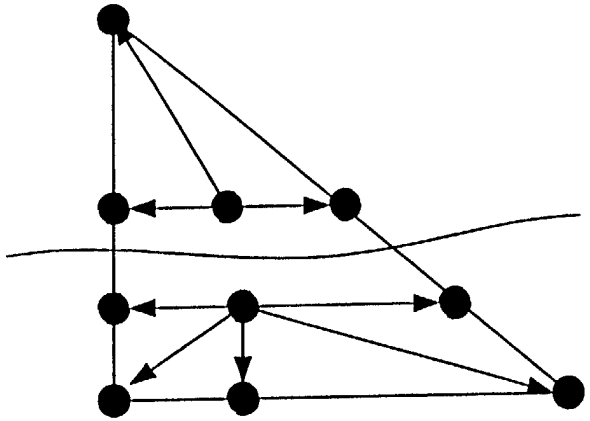

Figure 35

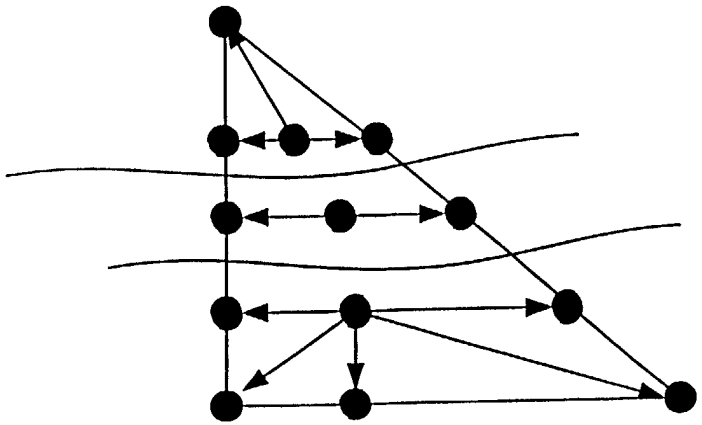

Figure 36

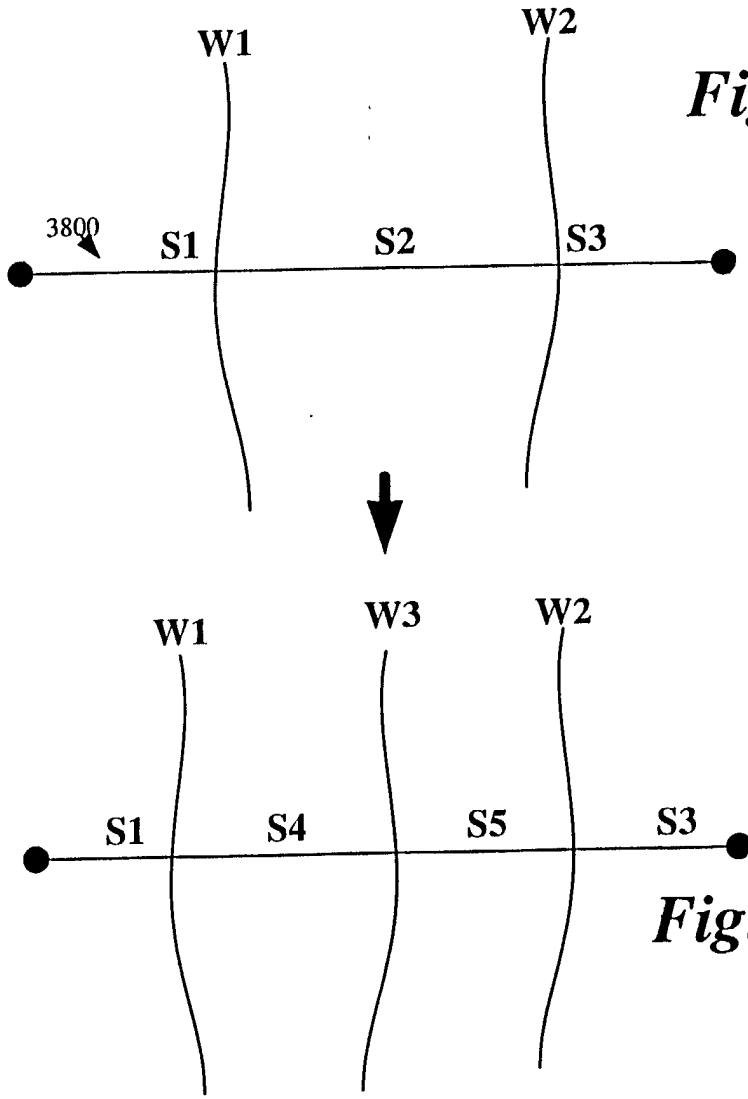

Figure 38A

Figure 38B

20020 "460990"

From: \ To:	Node	Face Item	Edge Item
Node	• Planarity • Vias	• Vias	• Planarity • Vias • Edge Capacity
Face Item	• Vias	• Vias	• Vias • Edge Capacity
Edge Item	• Planarity • Vias	• Vias	• Planarity • Vias • Edge Capacity

Figure 37

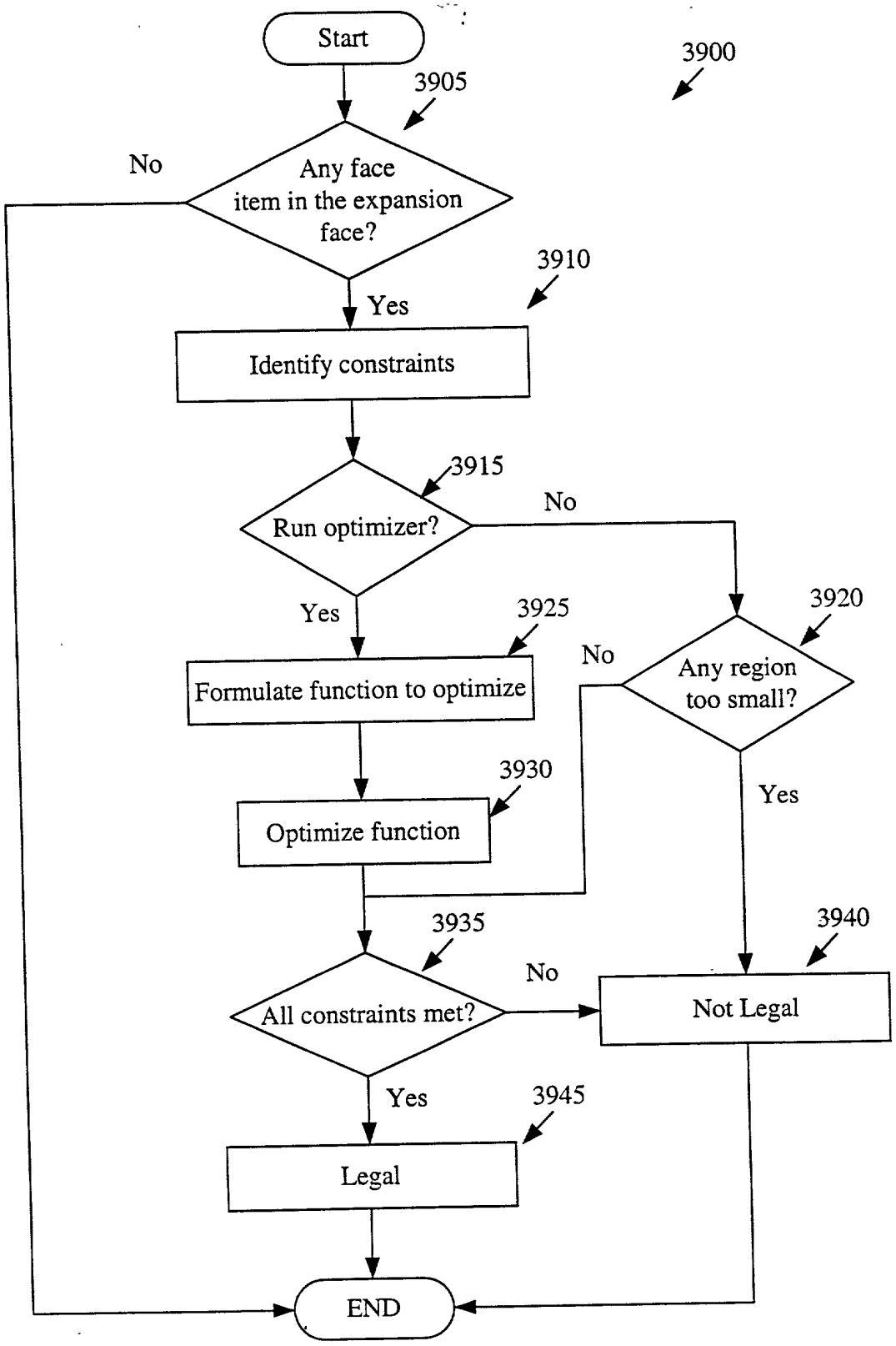
Start
3900
3905
Any face item in the expansion face?
No
Yes
3910
Identify constraints
3915
Run optimizer?
No
Yes
3925
Formulate function to optimize
3920
Any region too small?
No
Yes
3930
Optimize function
3935
All constraints met?
No
3940
Not Legal
Yes
3945
Legal
END

Figure 39A

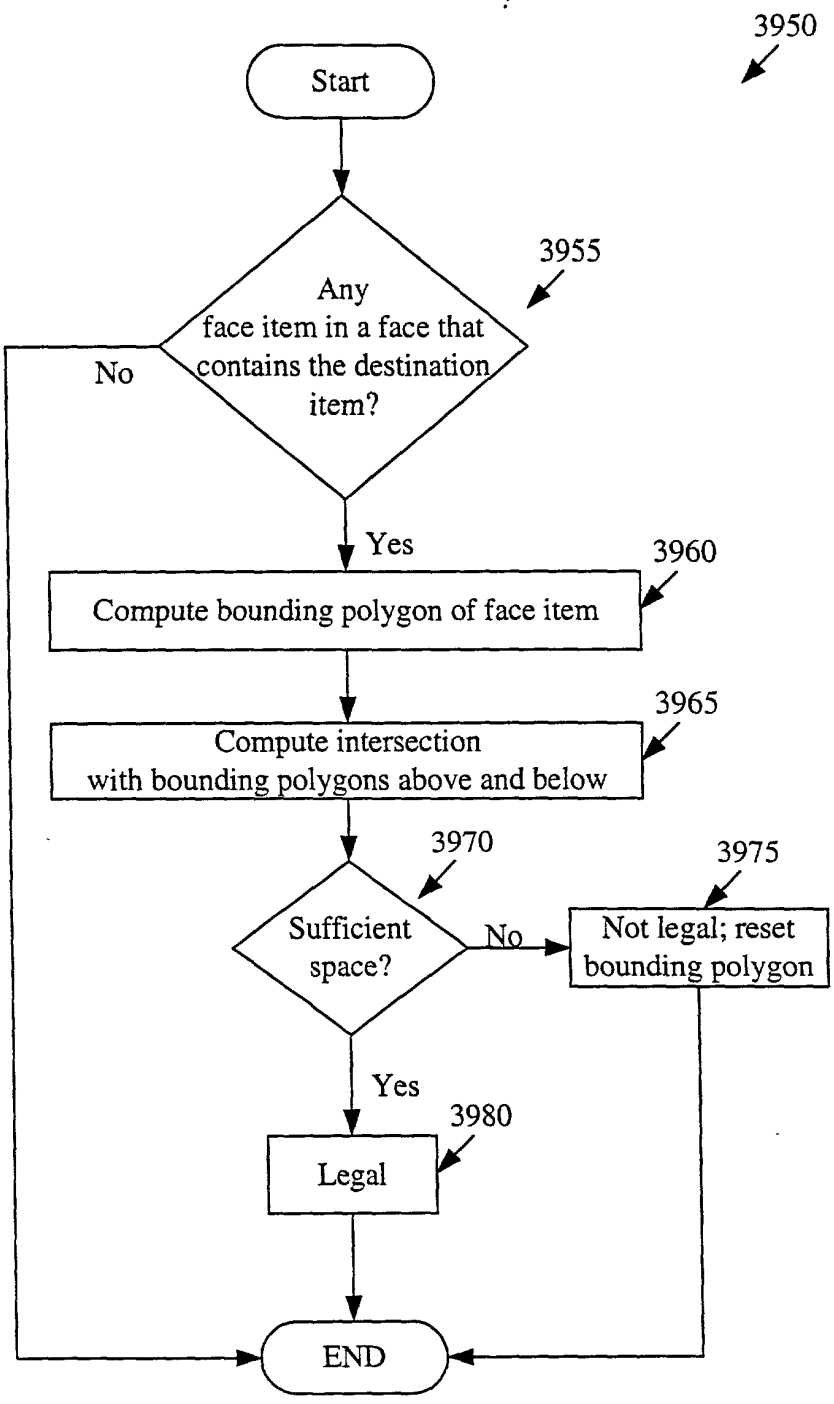
3950
Start
3955
Any face item in a face that contains the destination item?
No
Yes
3960
Compute bounding polygon of face item
3965
Compute intersection with bounding polygons above and below
3970
Sufficient space?
No
3975
Not legal; reset bounding polygon
Yes
3980
Legal
END

Figure 39B

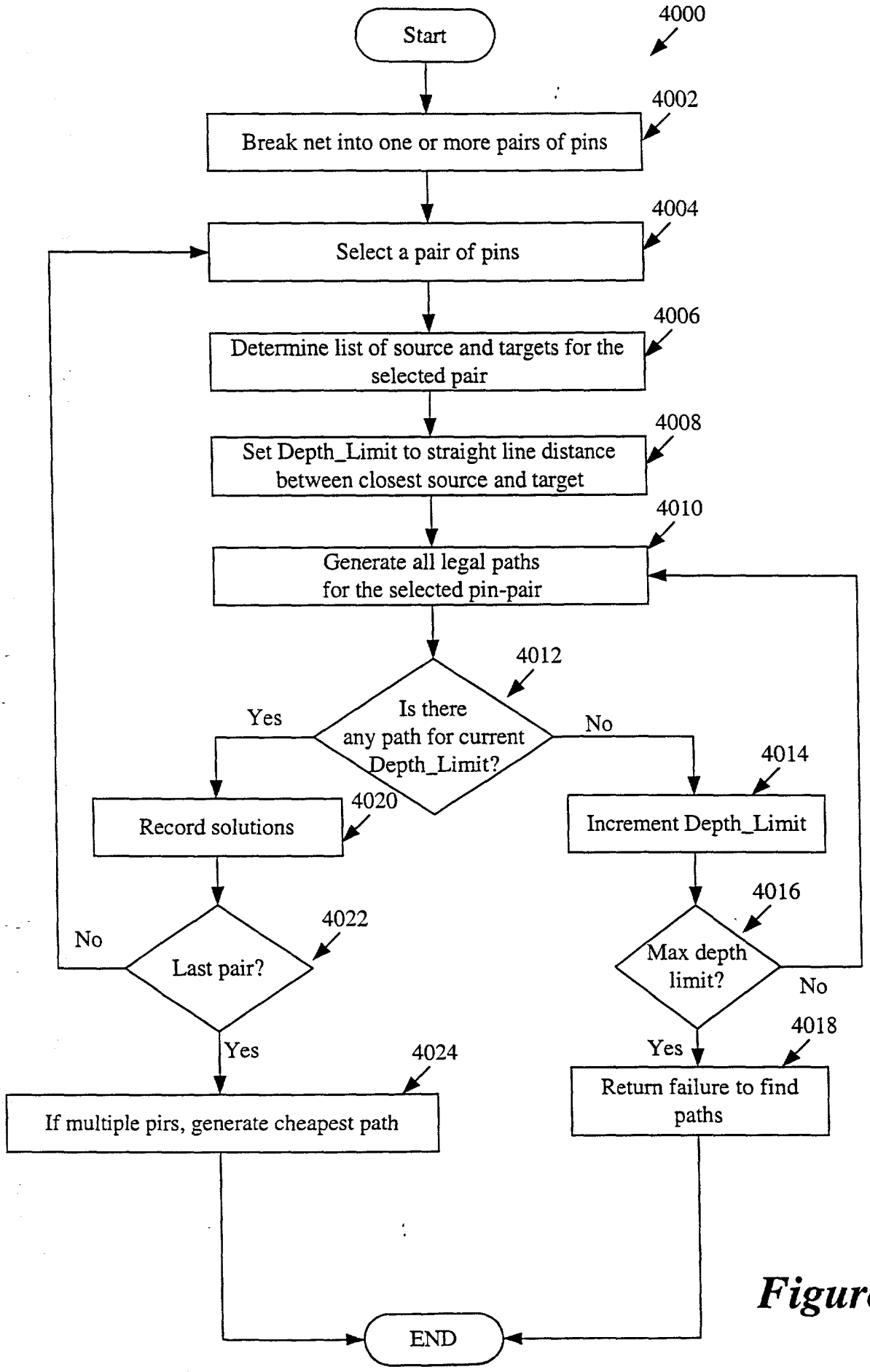
Start
4000
Break net into one or more pairs of pins
4002
Select a pair of pins
4004
Determine list of source and targets for the selected pair
4006
Set Depth_Limit to straight line distance between closest source and target
4008
Generate all legal paths for the selected pin-pair
4010
Is there any path for current Depth_Limit?
4012
Yes
No
Record solutions
4020
Increment Depth_Limit
4014
Last pair?
4022
No
Yes
Max depth limit?
4016
No
Yes
Return failure to find paths
4018
If multiple pirs, generate cheapest path
4024
END

Figure 40

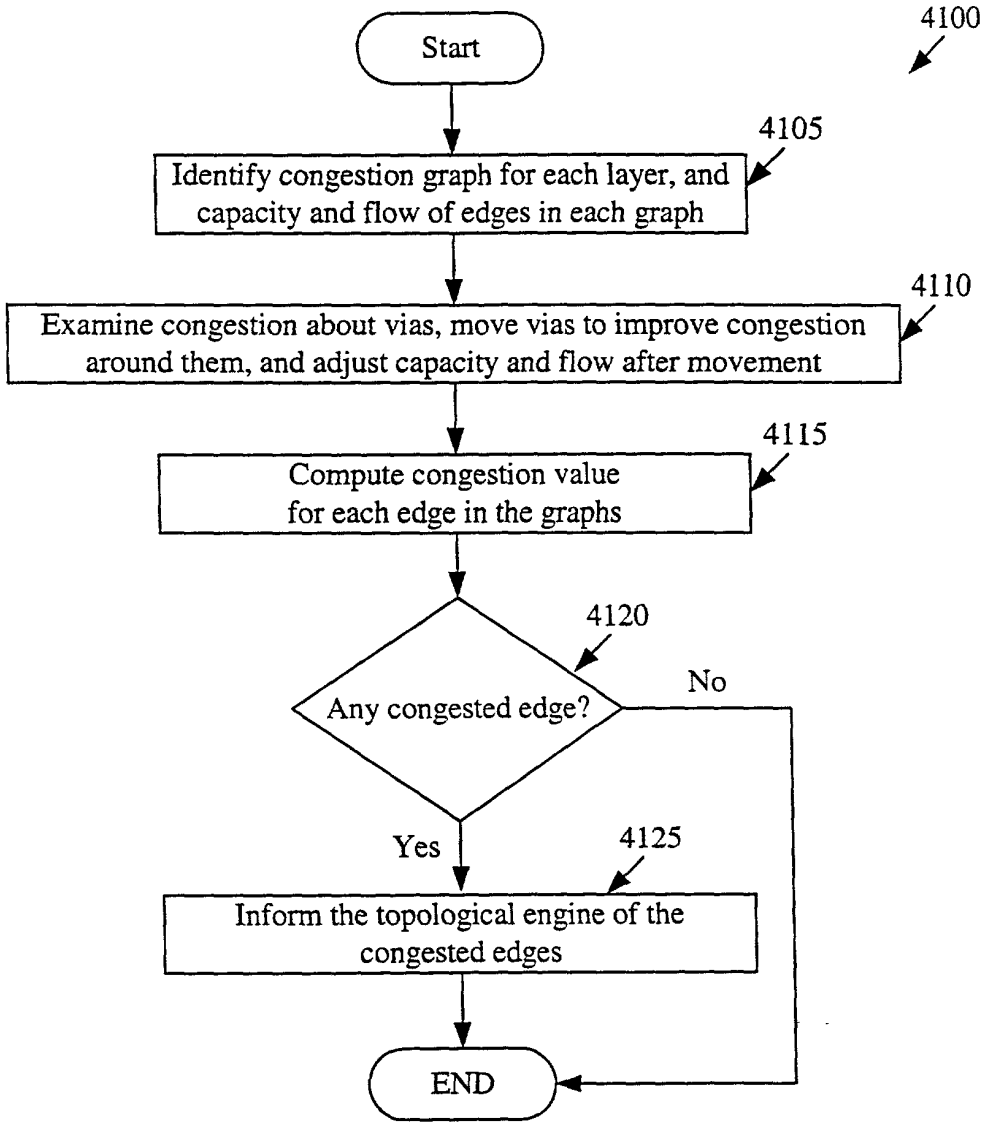
4100
Start
4105
Identify congestion graph for each layer, and capacity and flow of edges in each graph
4110
Examine congestion about vias, move vias to improve congestion around them, and adjust capacity and flow after movement
4115
Compute congestion value for each edge in the graphs
4120
Any congested edge?
No
Yes
4125
Inform the topological engine of the congested edges
END

Figure 41

20120704609900T

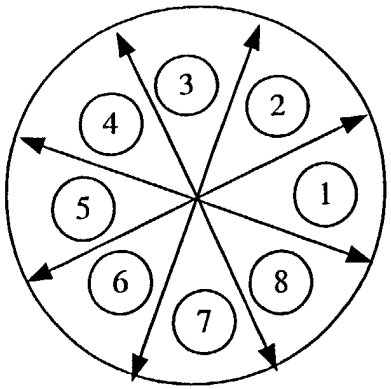

Figure 42

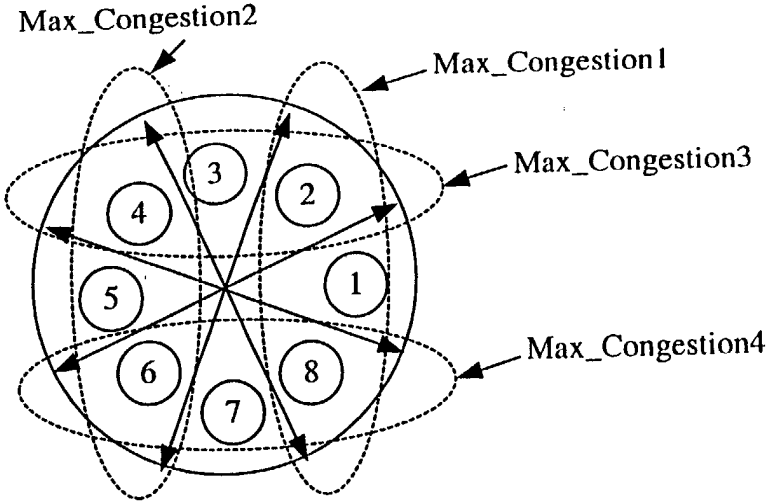

Figure 44

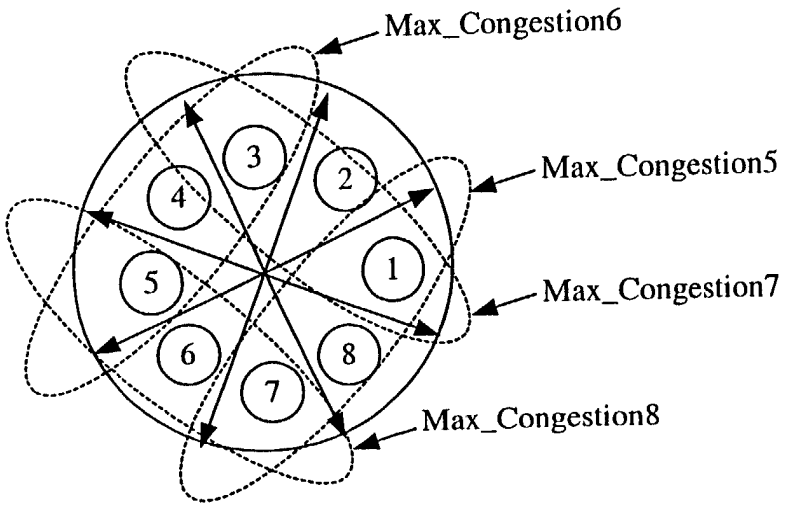

Figure 45

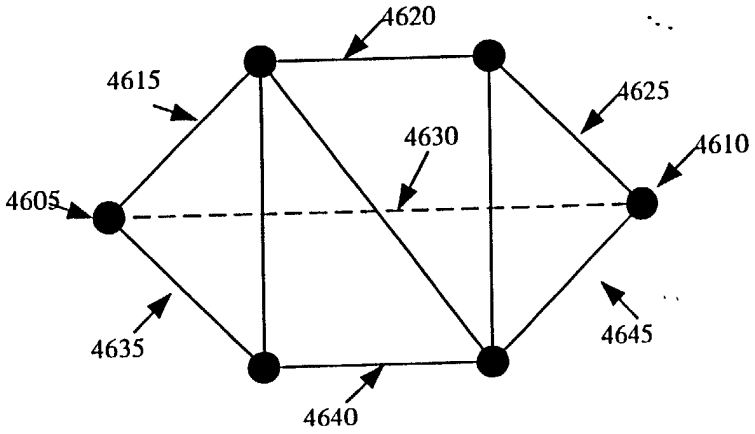

Figure 46

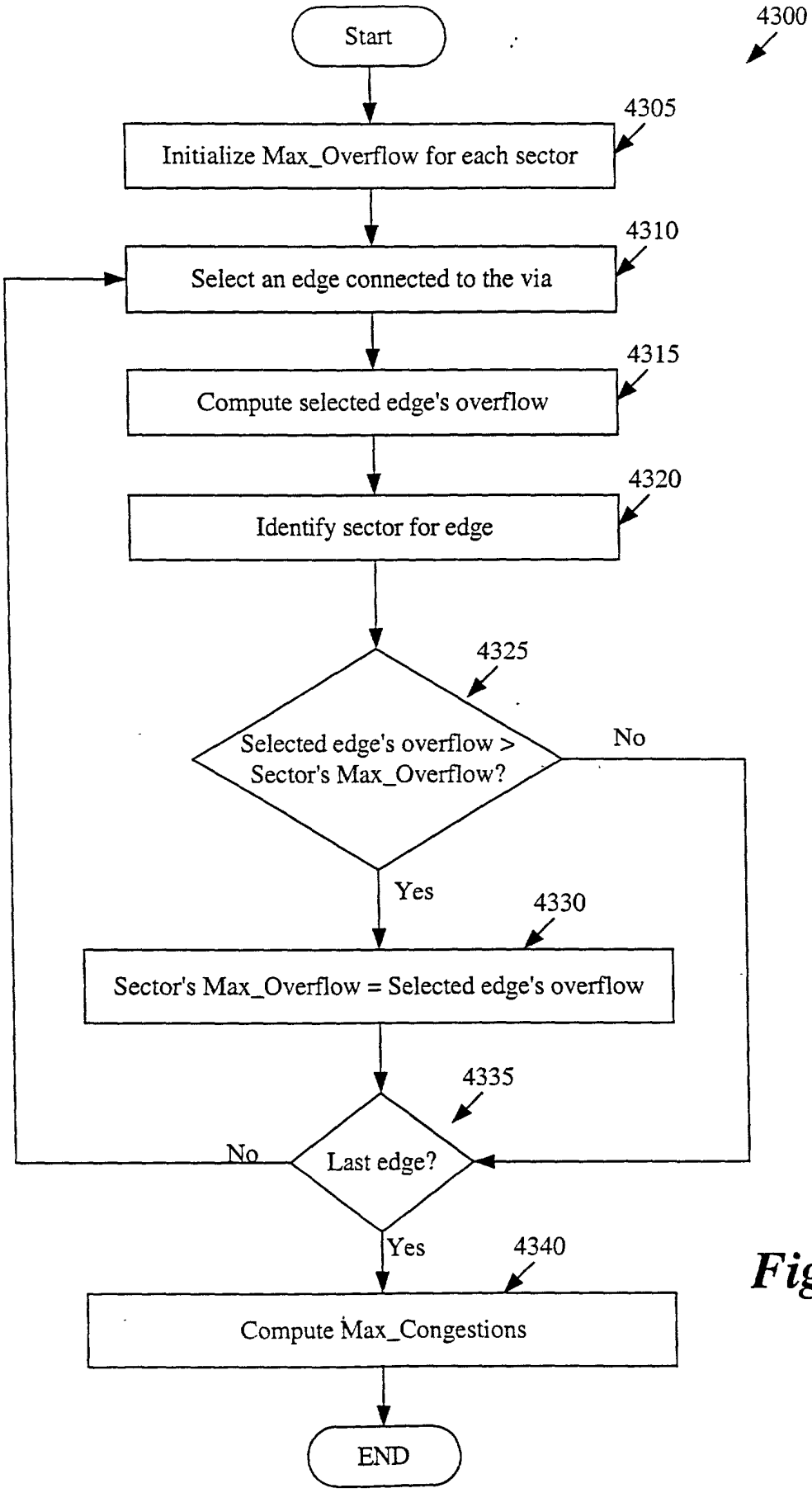
4300
Start
4305
Initialize Max_Overflow for each sector
4310
Select an edge connected to the via
4315
Compute selected edge's overflow
4320
Identify sector for edge
4325
Selected edge's overflow > Sector's Max_Overflow?
No
Yes
4330
Sector's Max_Overflow = Selected edge's overflow
4335
No
Last edge?
Yes
4340
Compute Max_Congestions
END

Figure 43

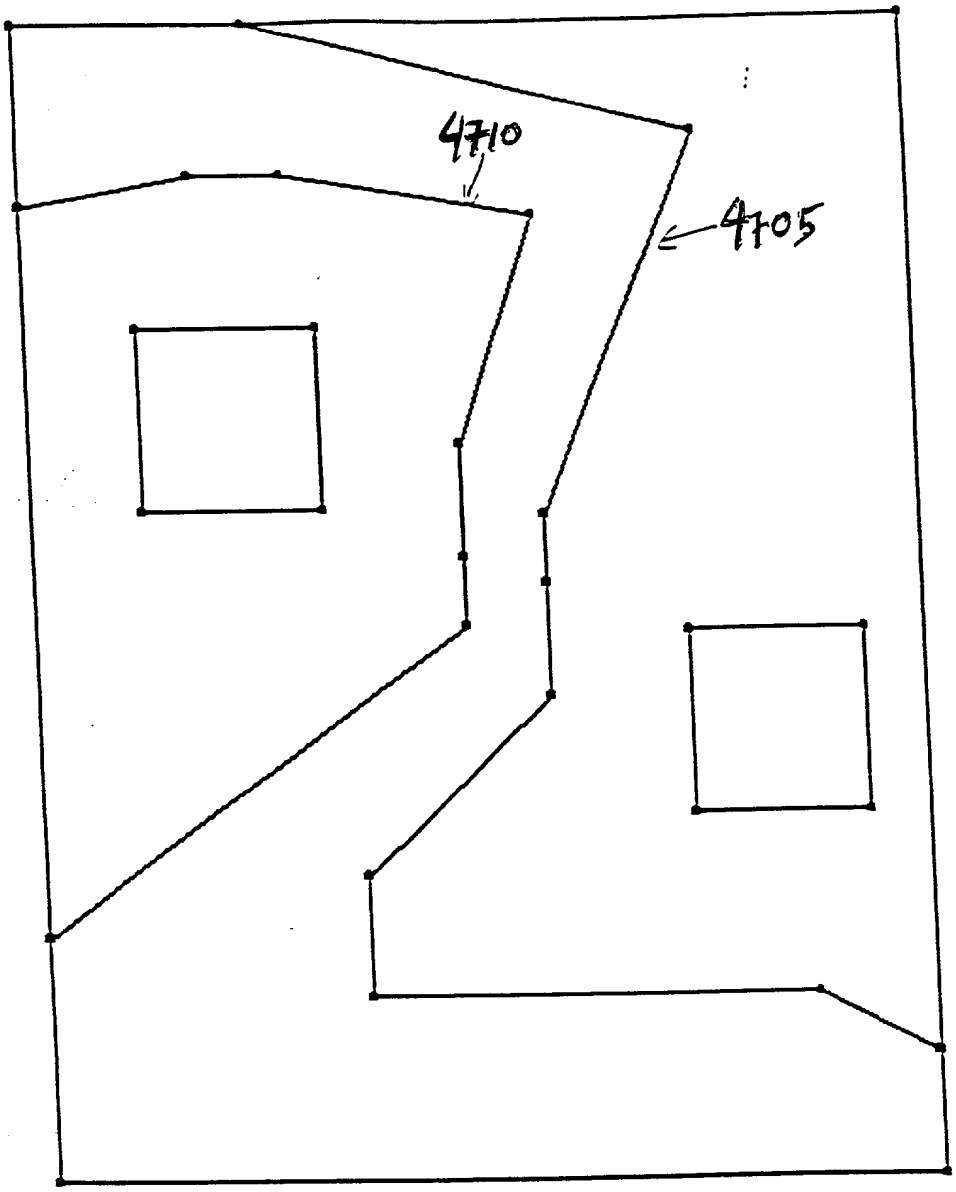

FIGURE 47

DRAFT 4/6/2004

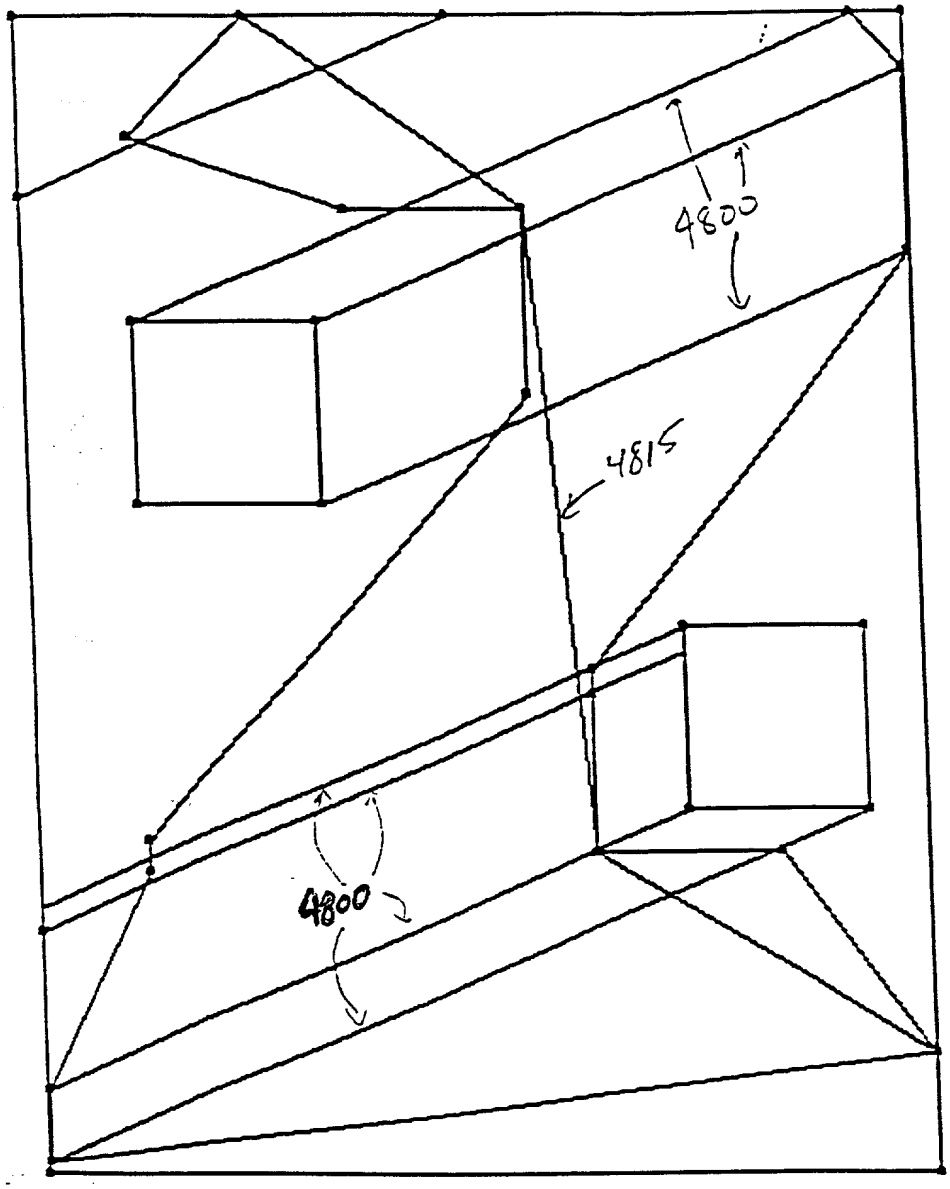

FIGURE 48A

DETECTED

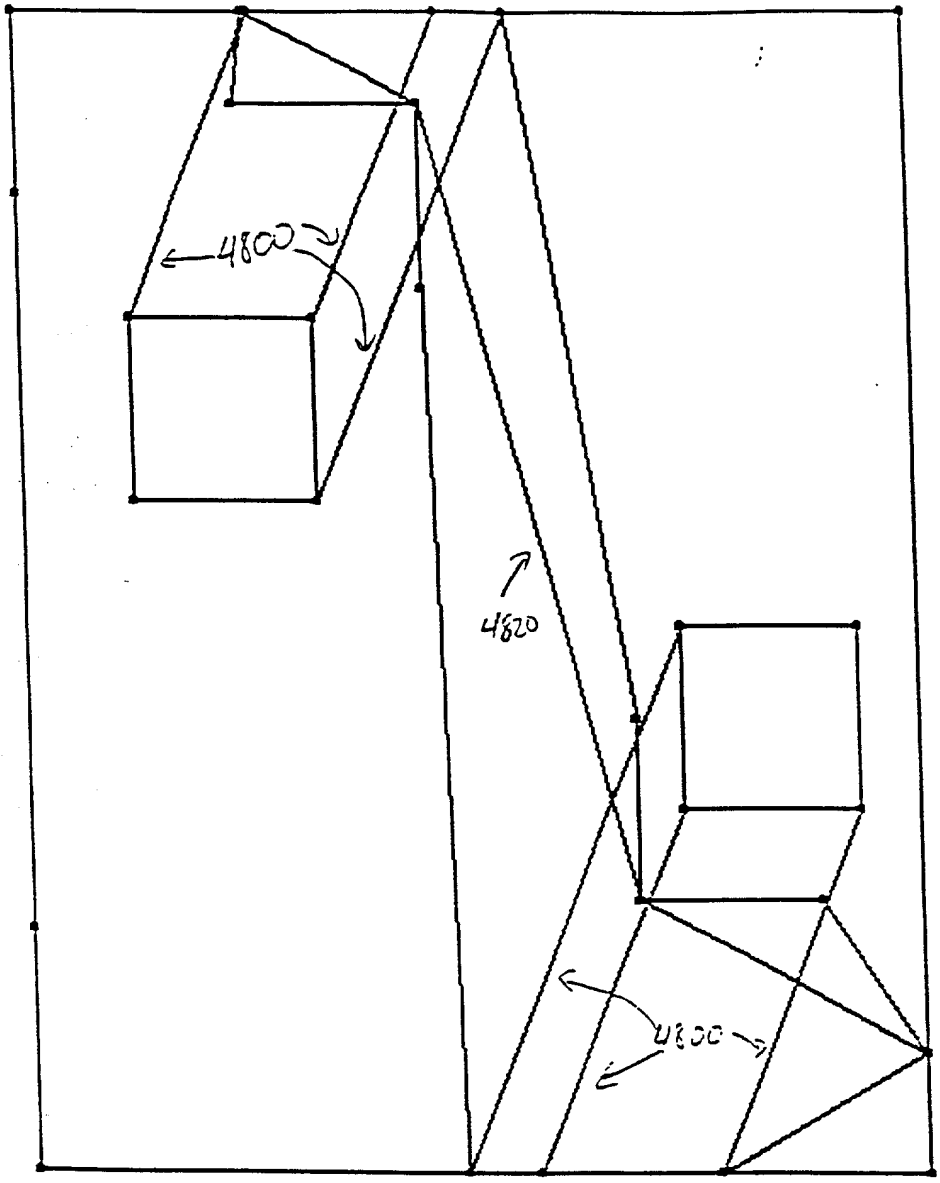

FIGURE 48B

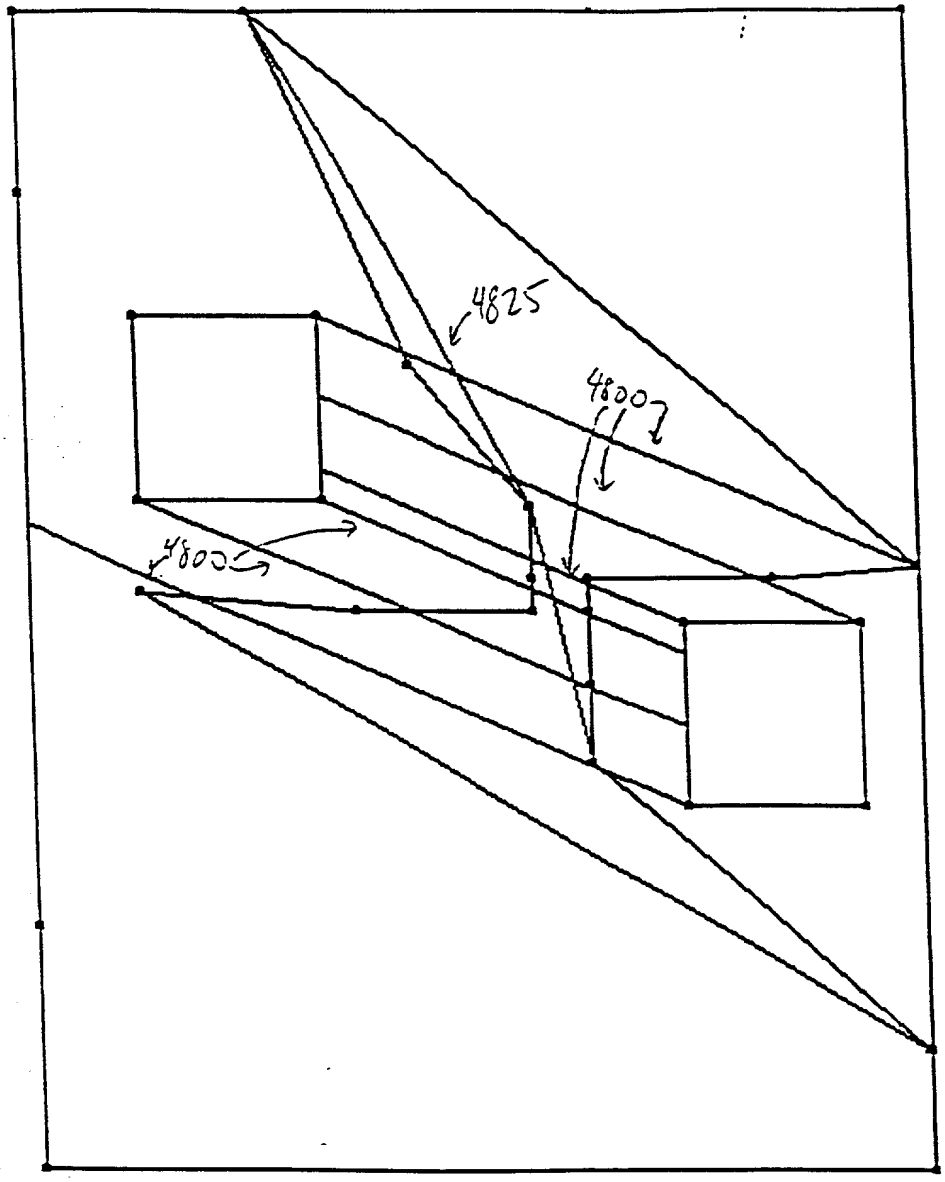

FIGURE 48C

CLIENT PROOF

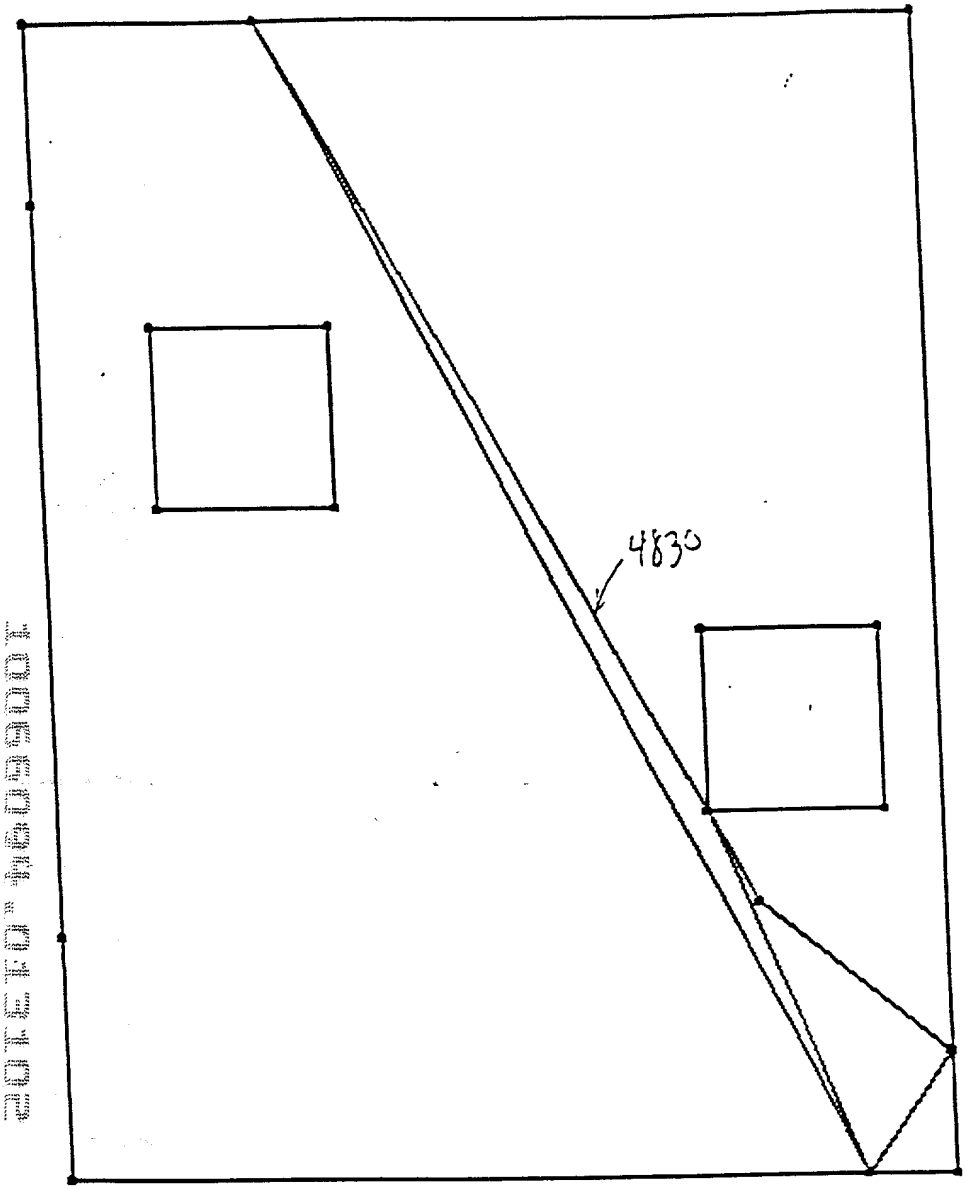

FIGURE 48D

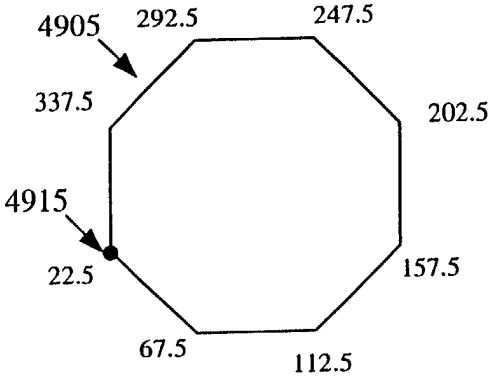

Figure 49A

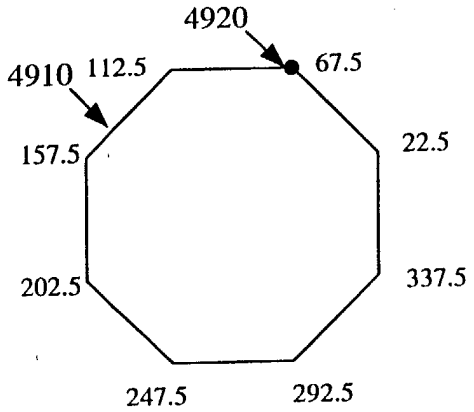

Figure 49B

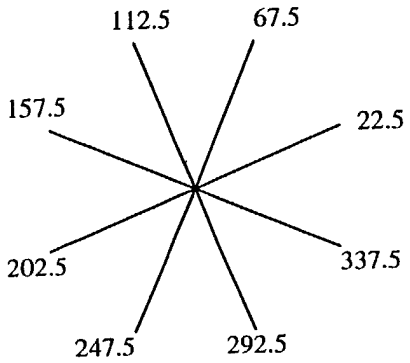

Figure 49C

SELECTED DRAWING

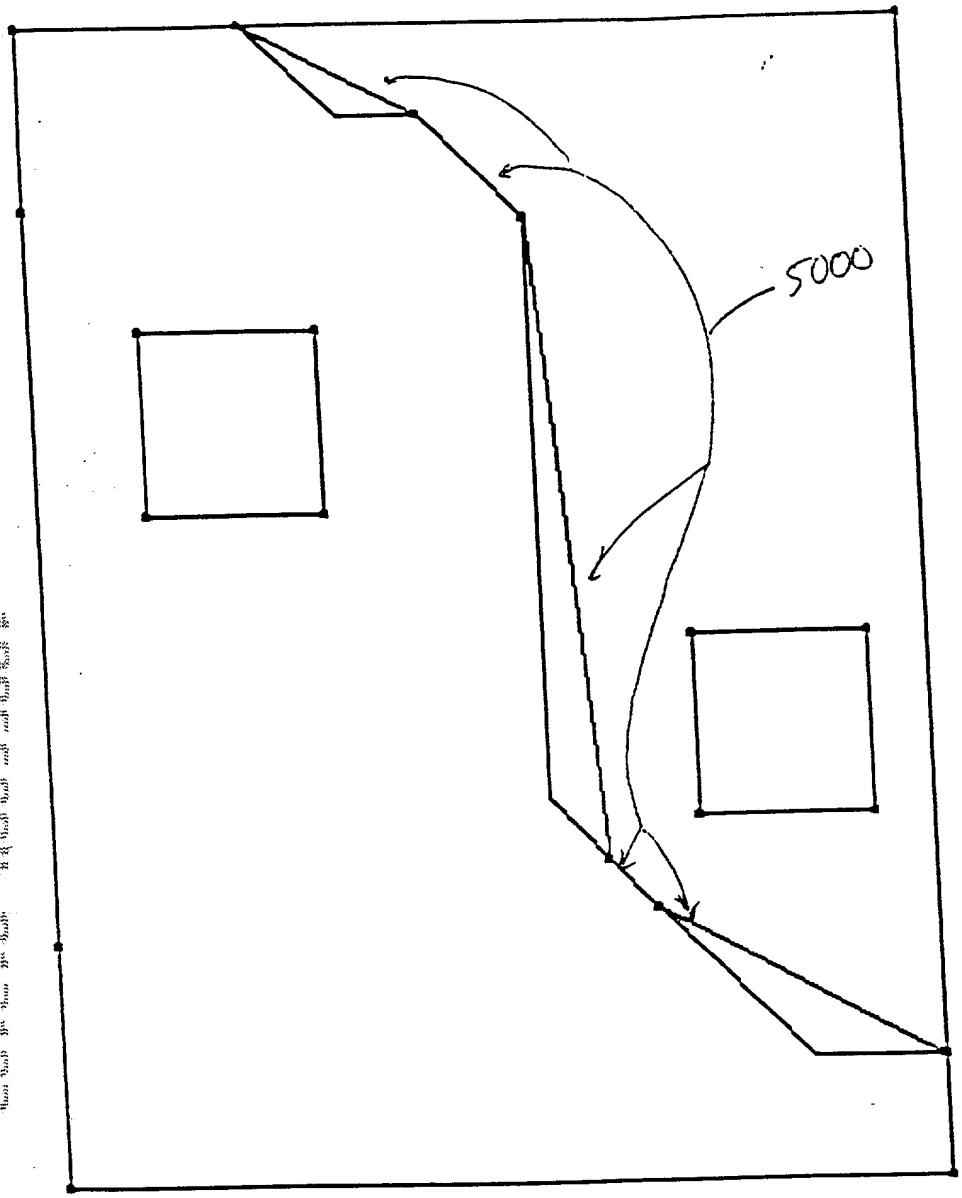

FIGURE 50

20150704609900A1

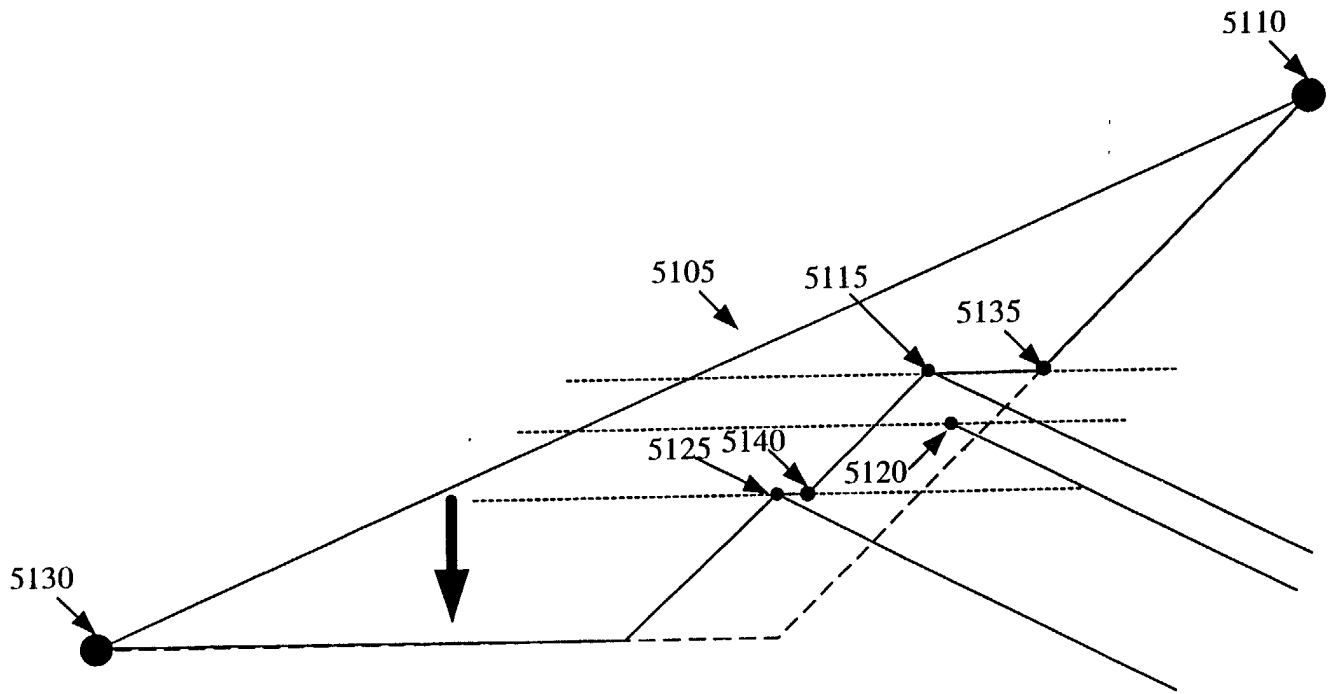

Figure 51

DO NOT REMOVE

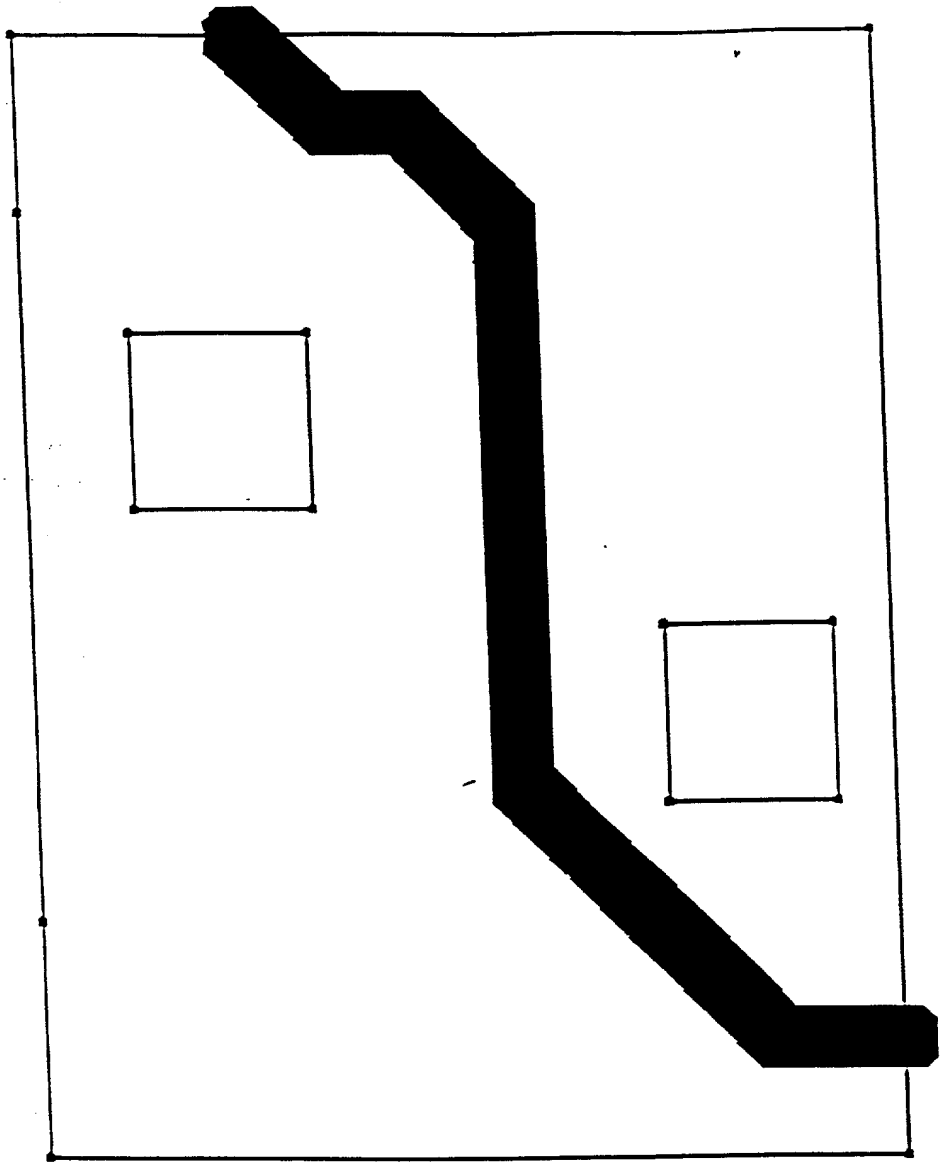

FIGURE 52

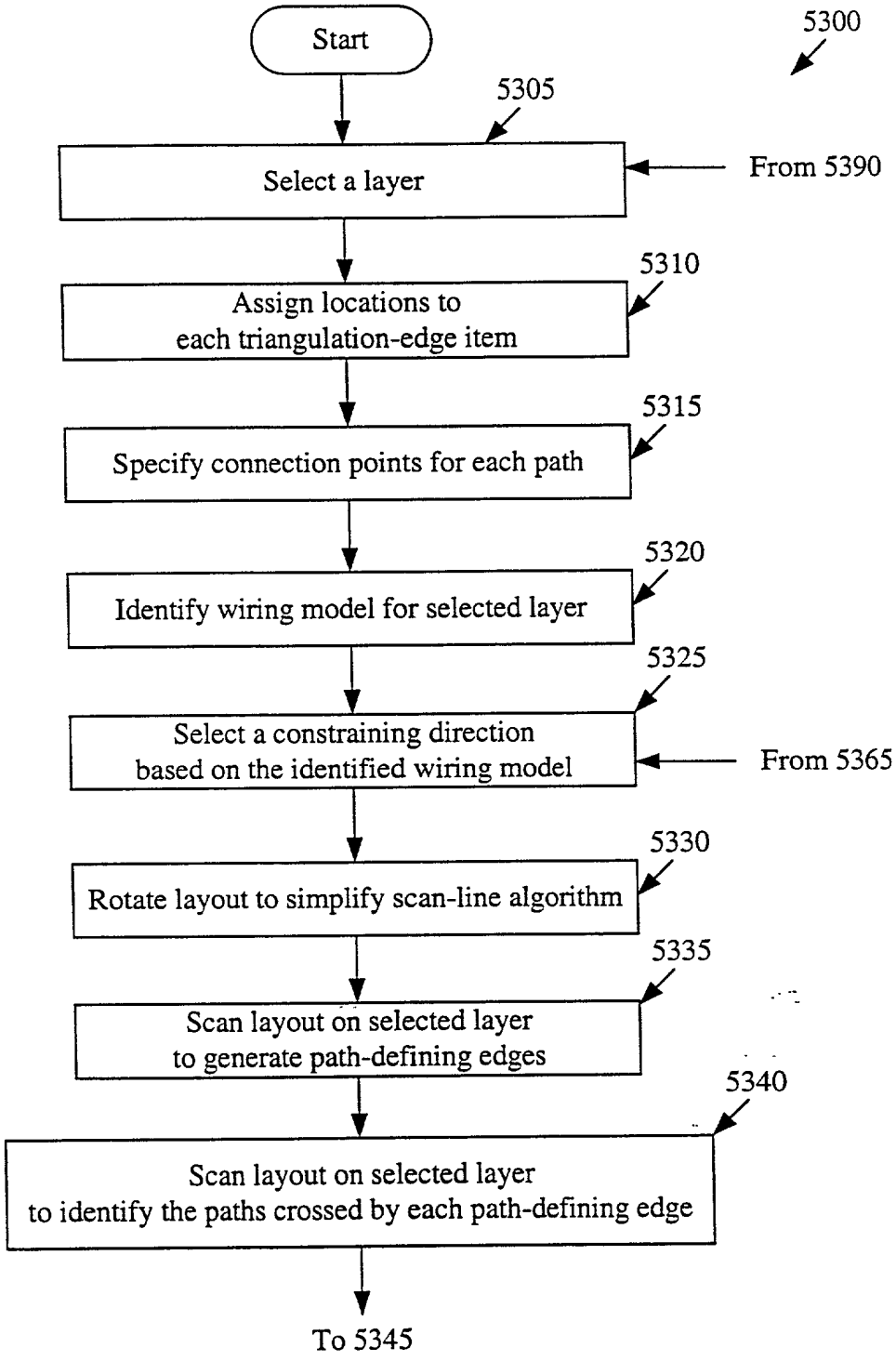

Figure 53

Figure 53: Figure 53A / Figure 53B

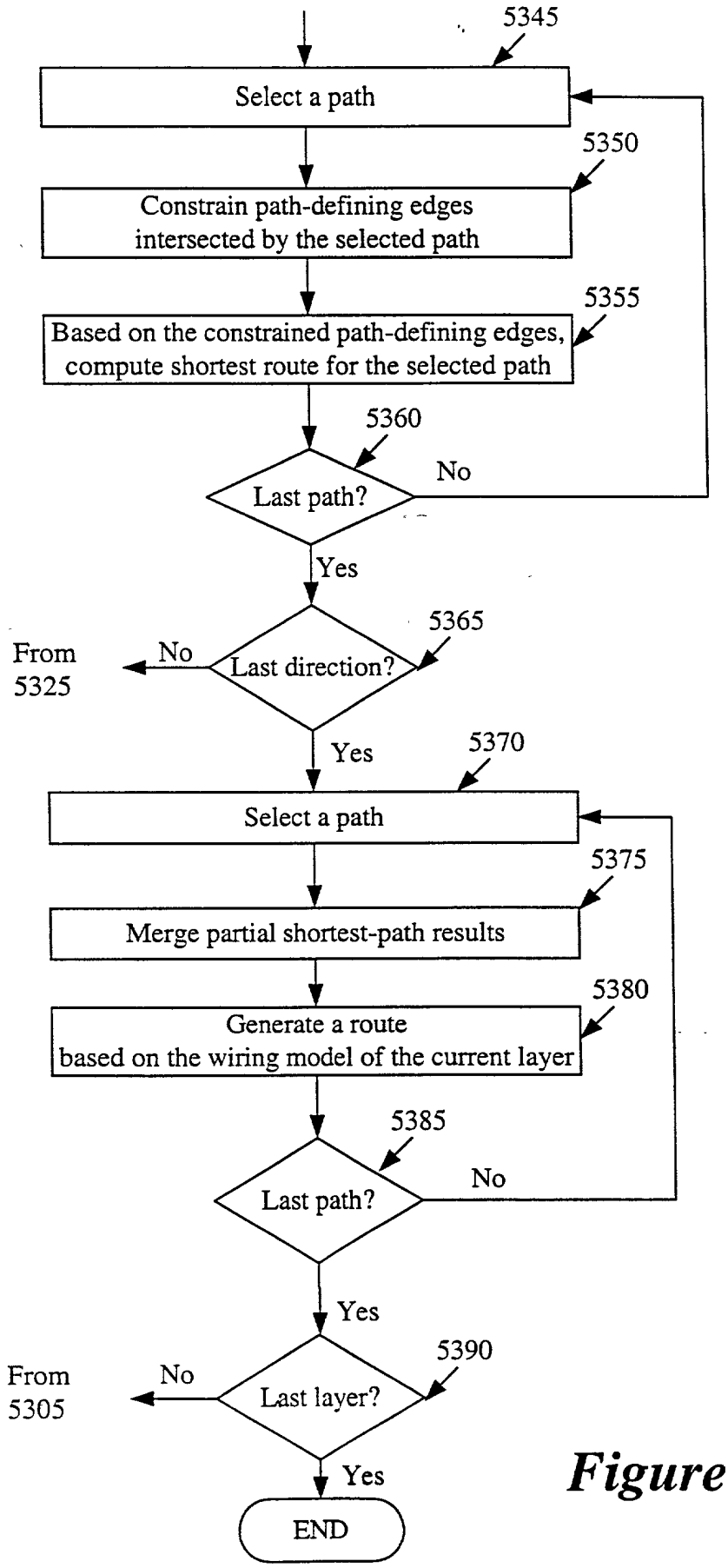
5345
Select a path
5350
Constrain path-defining edges intersected by the selected path
5355
Based on the constrained path-defining edges, compute shortest route for the selected path
5360
Last path?
No
Yes
5365
From 5325
No
Last direction?
Yes
5370
Select a path
5375
Merge partial shortest-path results
5380
Generate a route based on the wiring model of the current layer
5385
Last path?
No
Yes
5390
From 5305
No
Last layer?
Yes
END

Figure 53B

20TETO "W6099001

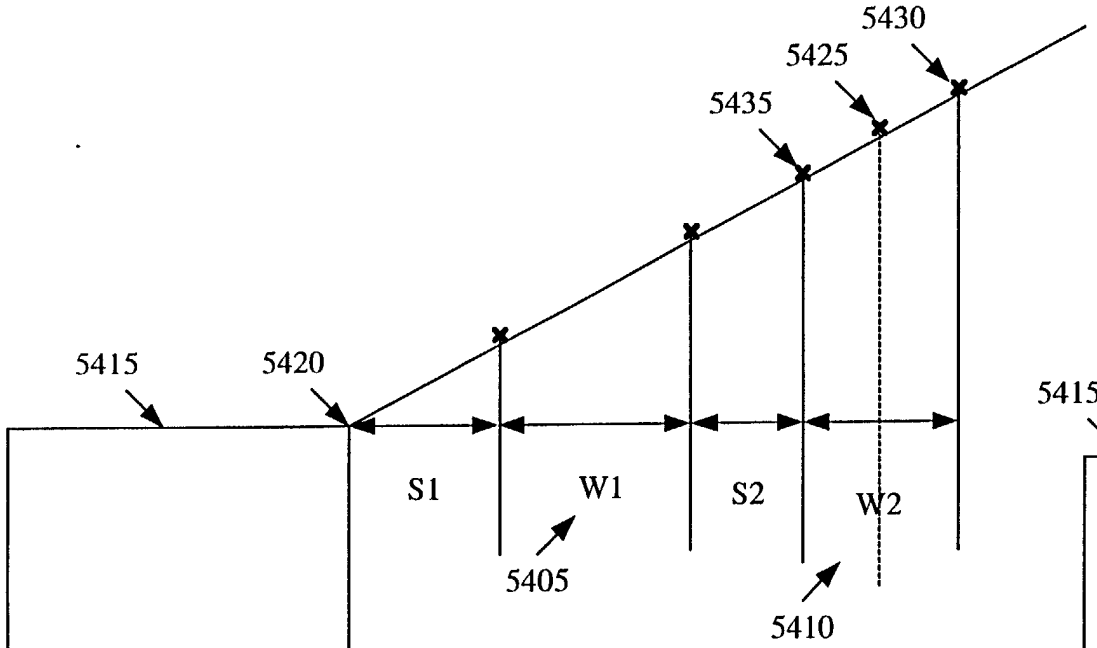

Figure 54

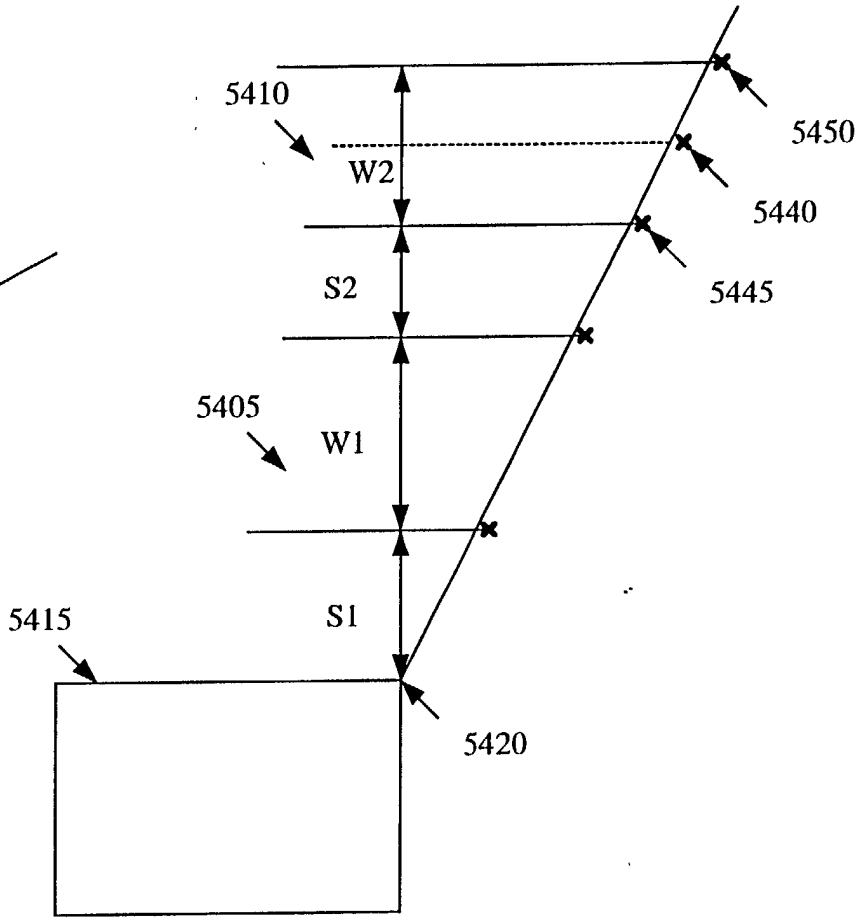

Figure 55

20TETD"h6099QQT

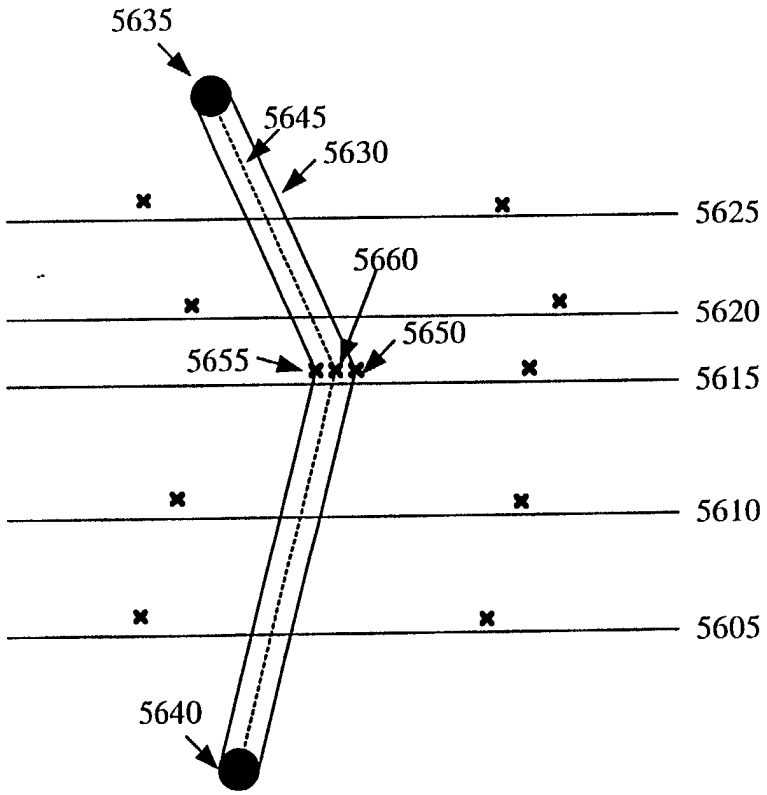

Figure 56

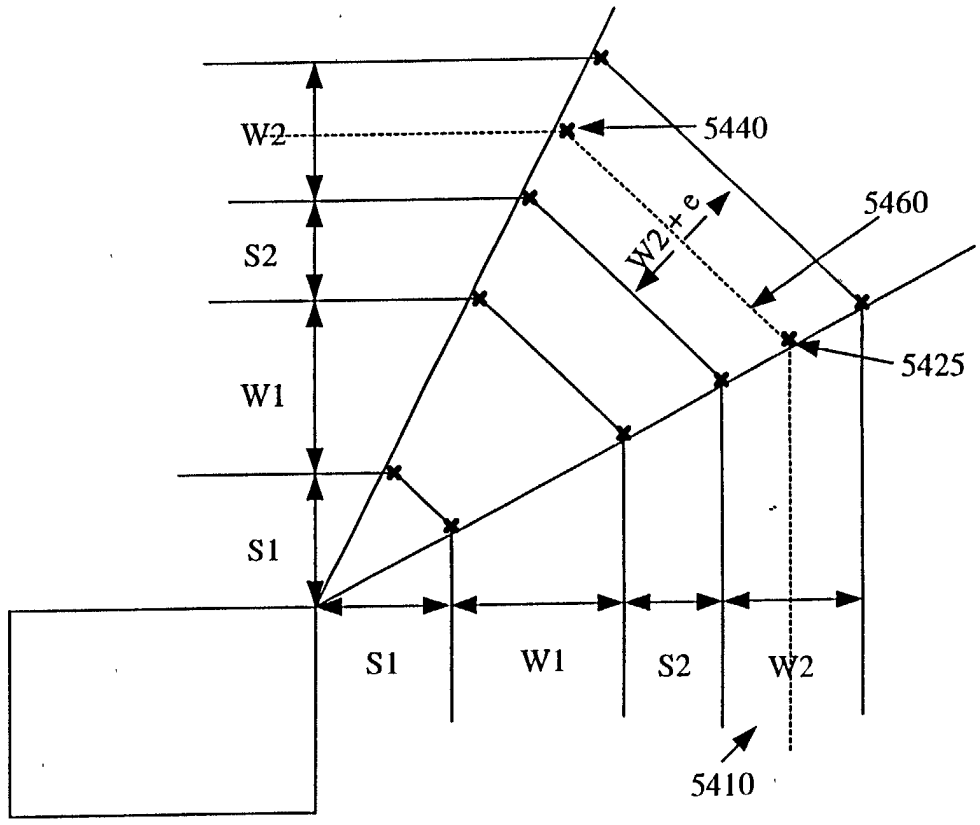

Figure 57

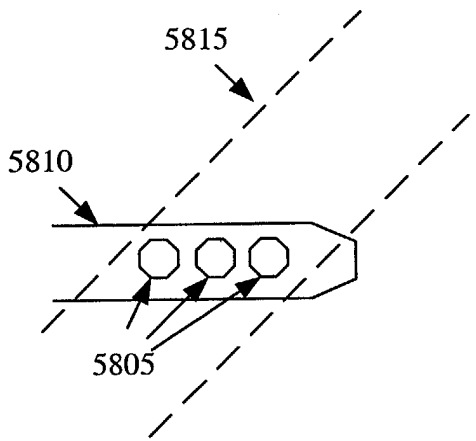

Figure 58

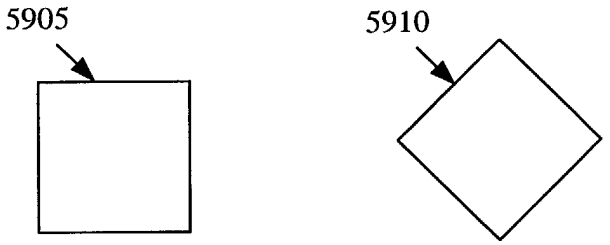

Figure 59

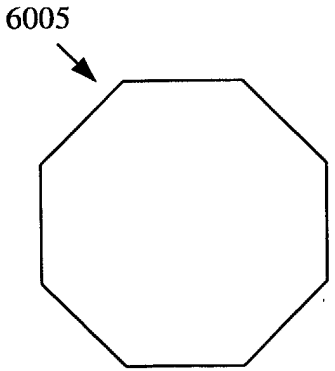

Figure 60

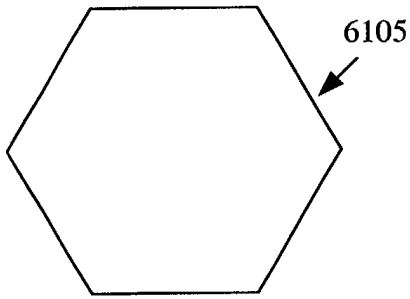

Figure 61

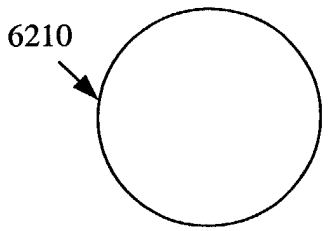

Figure 62

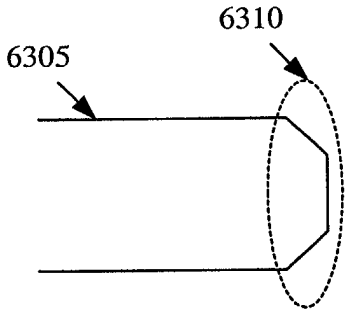

Figure 63

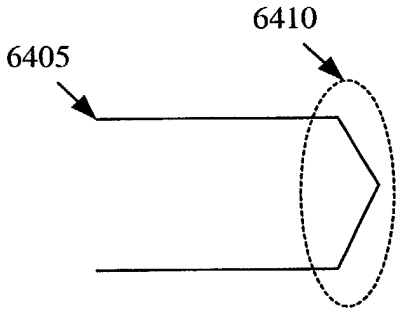

Figure 64

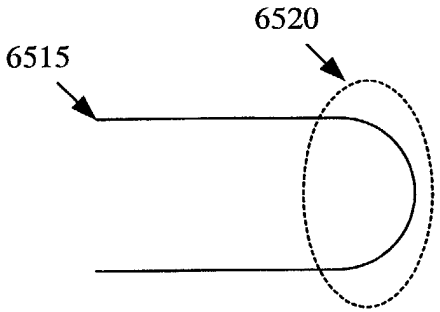

Figure 65

(1) *(2)* *(3)*

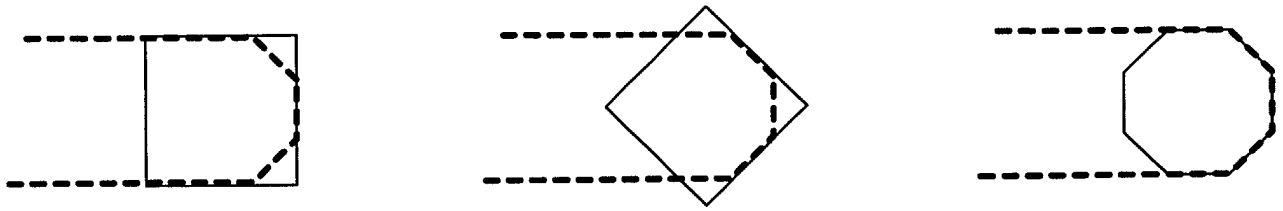

Figure 66

Figure 67

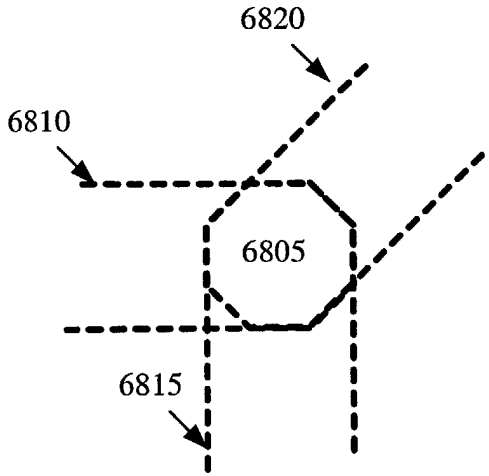

Figure 68

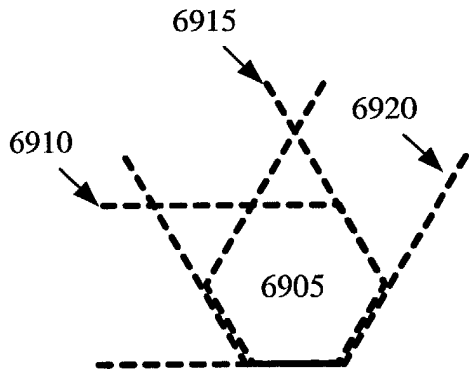

Figure 69

20150 4609907

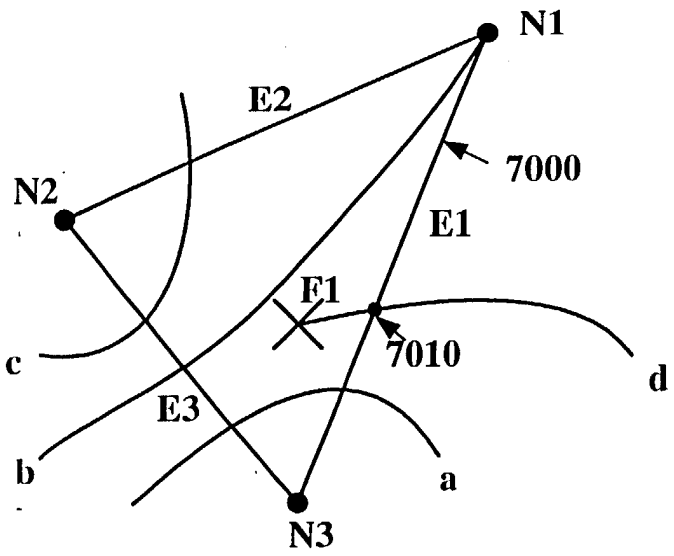

Figure 70

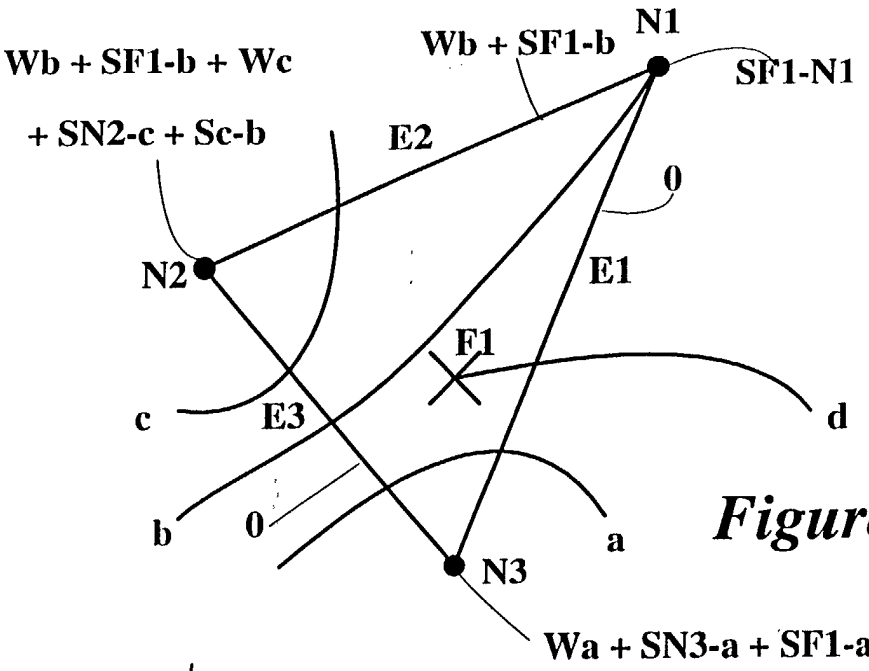

Figure 71

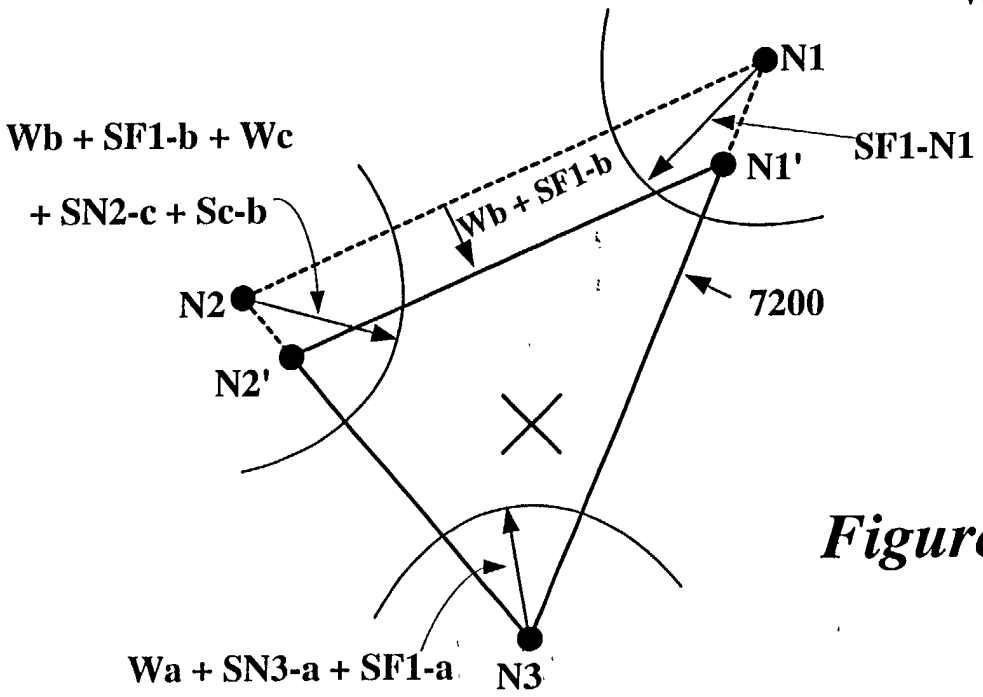

Figure 72

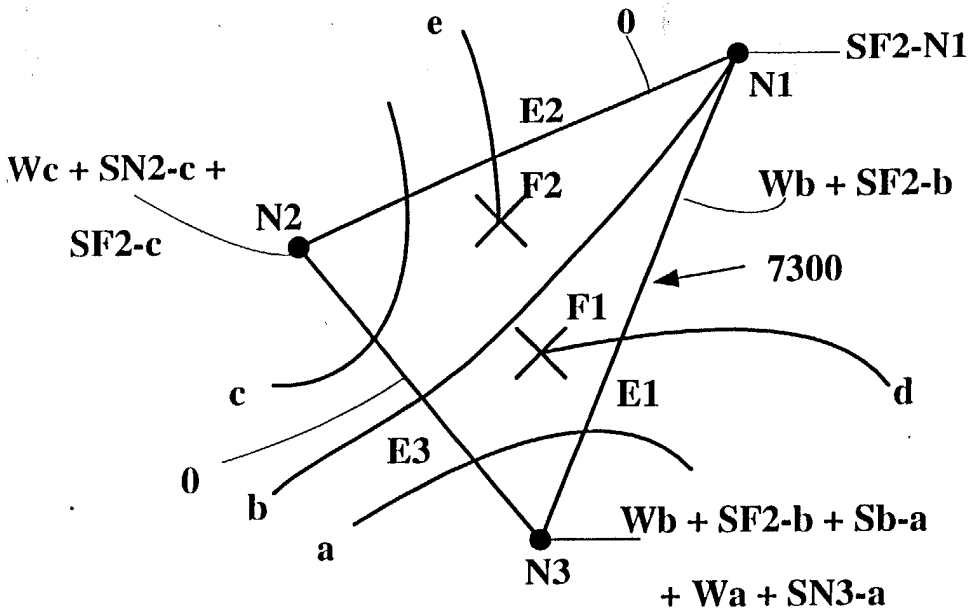

Figure 73

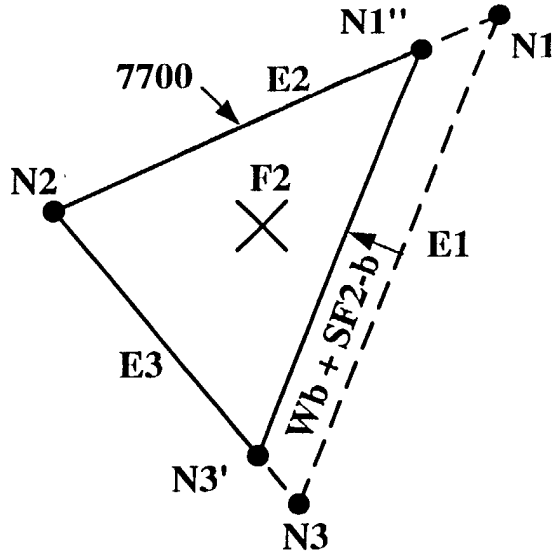

Figure 77

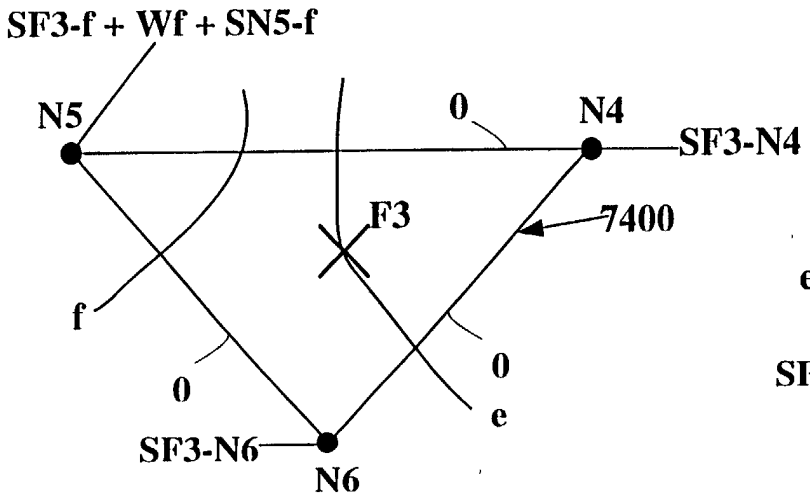

Figure 74

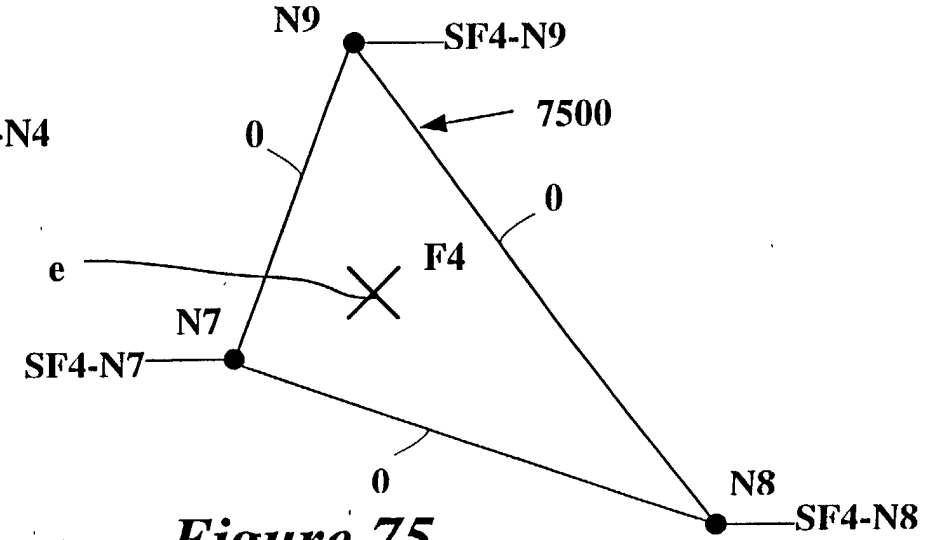

Figure 75

20TE10 H6099001

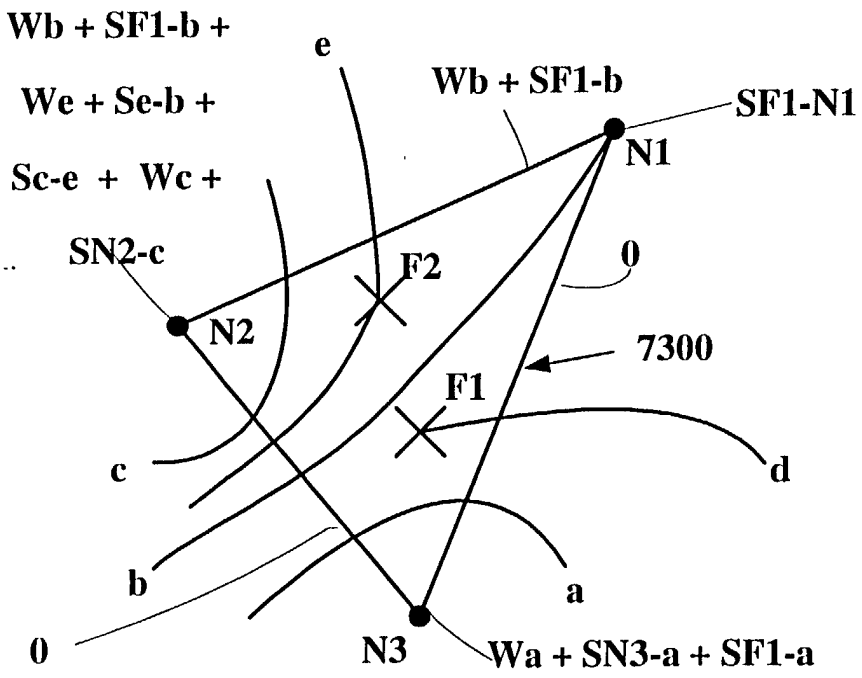

Figure 76

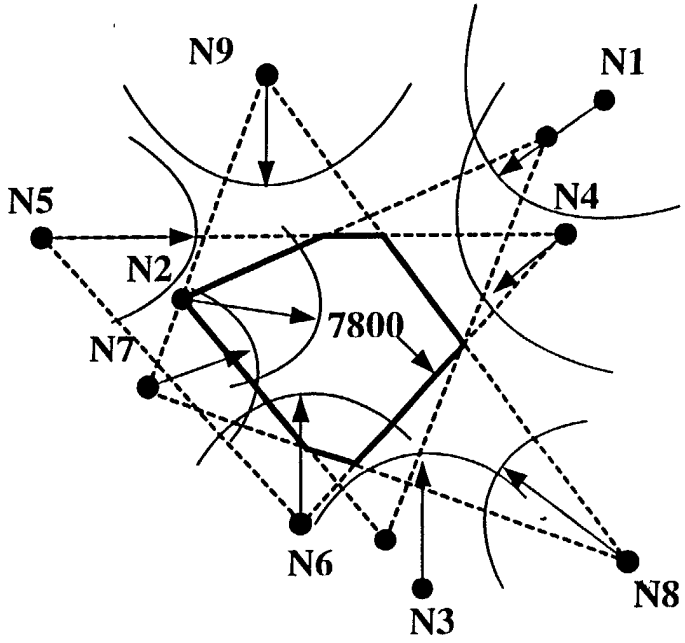

Figure 78

20TETO"1609900T

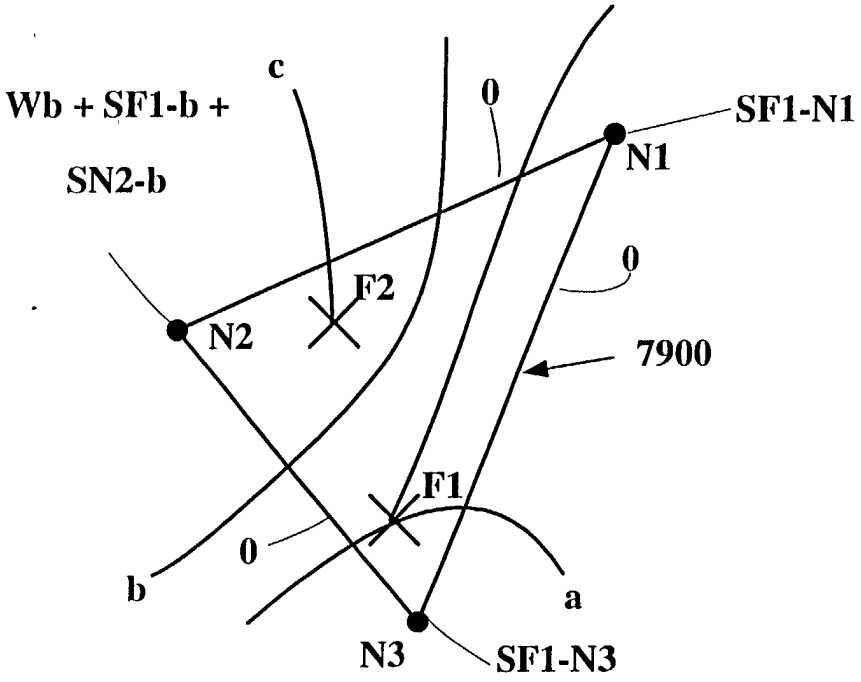

Figure 79

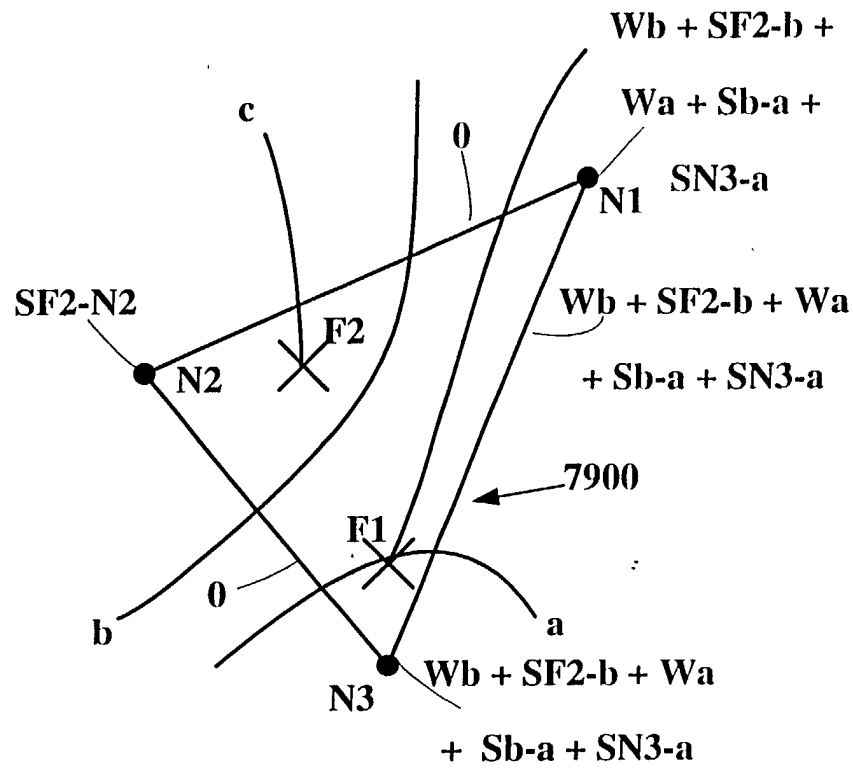

Figure 80

20150704609900T

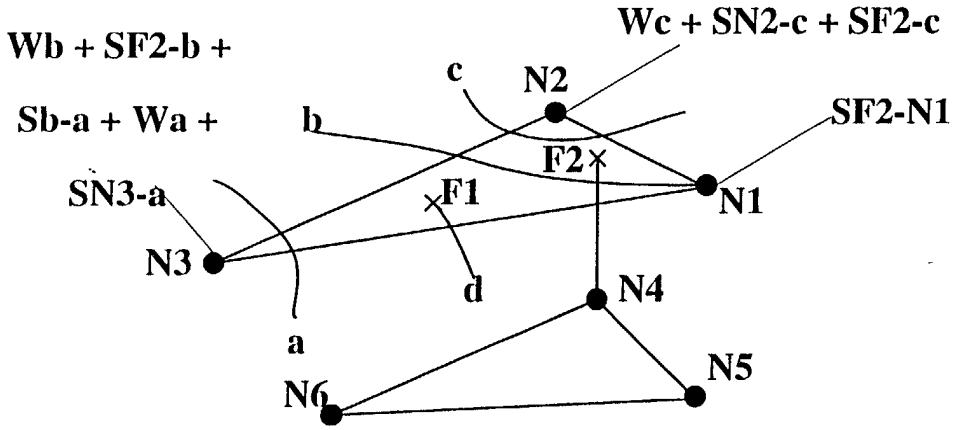

Figure 81

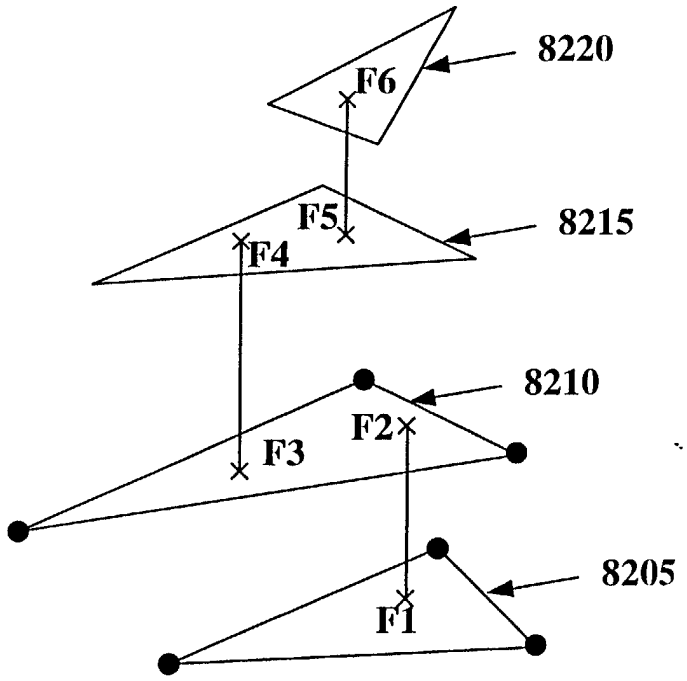

Figure 82

20TETO"4609900T

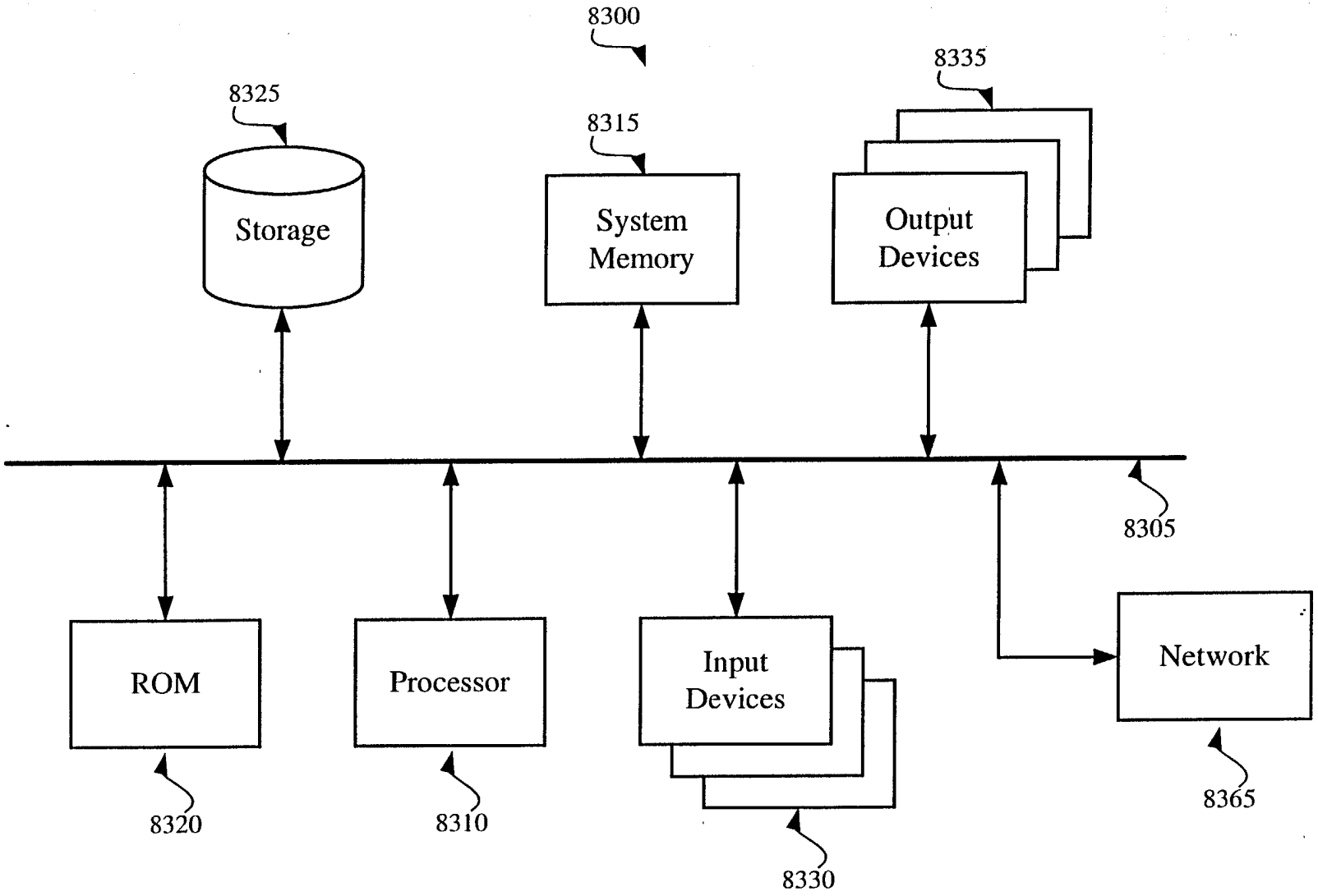

Figure 83